DOE/NASA TECHNICAL MEMORANDUM

DOE/NASA TM-78251

A COMPUTER PROGRAM FOR ASSESSING THE ECONOMIC FEASIBILITY OF SOLAR ENERGY FOR SINGLE-FAMILY RESIDENCES AND LIGHT COMMERCIAL APPLICATIONS

By J. Alan Forney
Cost Analysis Office
Systems Analysis and Integration Laboratory

and

David Walker and Mike Lanier
Computer Sciences Corporation
Huntsville, Alabama

September 1979

For the U. S. Department of Energy

U.S. Department of Energy

Solar Energy

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

TECHNICAL MEMORANDUM

A COMPUTER PROGRAM FOR ASSESSING THE ECONOMIC FEASIBILITY OF SOLAR ENERGY FOR SINGLE-FAMILY RESIDENCES AND LIGHT COMMERCIAL APPLICATIONS

I. INTRODUCTION

A solar heating system operating on free sunlight can typically supply 50 to 70 percent of the home space heating and hot water load. This savings in conventional fuel cost (either electricity, natural gas or oil) over many years in the future is the incentive for spending the extra money to install a solar heating system. Because of the possibility of extended bad weather, a conventional heating system capable of providing 100 percent of the heating load is always required. This conventional system would serve as a backup when a solar heating system is also installed. So the question of whether to spend the extra money depends on how much you can expect to save in the future and, perhaps more importantly, how long you are willing to wait to get those savings.

Classically, life cycle costing is the method used to evaluate economic alternatives. The individual cash flow profiles of each alternative are discounted to present dollars and the cheapest selected. However, individual homeowners do not generally make decisions on a life cycle cost basis. Fifteen or twenty years is probably too long to wait and the first four or five years is when most people would want their investment in solar to "pay back." The following definitions, from Reference 1, are useful:

> For individual homeowners, solar marketers have found that cash flow measures are paramount. Three measures are particularly important, each in terms of time:
>
> - Years to positive cash flow is the number of years for fuel cost savings to become greater than the extra expenses of the solar system, after taxes. The criterion assumed is two years, because most homeowners expect rapid savings for an "energy conserving" investment.
>
> - Years to recover downpayment is the number of years required for accumulated savings to offset initial cash payments and early cash flow losses, after taxes. The criterion assumed is five years, based on today's

average housing turnover rate of five years and the expectation that homeowners will seek to recover their downpayment before selling.

- Payback period is the number of years for accumulated savings to repay the full cost of the system (or equal the remaining principal on a loan if financed). The criterion assumed is ten years, assuming the homeowner wishes to be paid back entirely for his investment in no more than ten years.

The computer program provides a cash flow analysis for each year so that any decision criteria selected can be applied with the data provided.

There are some important economic considerations involved in "sizing" a solar heating system. Here "sizing" means selecting a collector area and performance and determining the resultant percentage of the total annual load carried by the solar system. See Reference 2 for a more detailed discussion. There are fixed and variable costs in a solar installation. Fixed costs consist of pumps, controls, ducts, etc. Variable costs are mainly due to collector area. As the collector area increases, the solar system carries a greater percentage of the annual load. The characteristic curve of solar system life cycle cost versus collector area is depicted in Figure 1. For low collector areas, not much sunlight is captured; hence, little heating value is achieved. A relatively large amount of money has been spent to perform the heating equivalent of a small amount of conventional fuel. At high collector areas a similar uneconomical phenomena occurs. Adding collector area, when it is already relatively high, provides a small increase in the percent of total load carried by the solar system. Hence the added collector area and cost is only slightly reducing the conventional fuel bill. There is a collector area which minimizes life cycle cost, as depicted on Figure 1. However, this may not be the size of solar system most attractive to the homeowner from an economic standpoint. Consider the economic decision involved in moving the design from the area of minimum life cycle cost along Path A (i.e., in the direction of reducing the collector area). The characteristic curve is relatively flat near the area of minimum life cycle cost. Reducing collector area has an almost linear effect on reducing system initial cost. A penalty is paid, however, in terms of increased life cycle cost. The important point here is that savings in initial cost may be relatively large, and hence attractive to the homeowner, in comparison to the life cycle cost penalty. The sensitivity study and plot features of the computer program allow these alternatives to be studied.

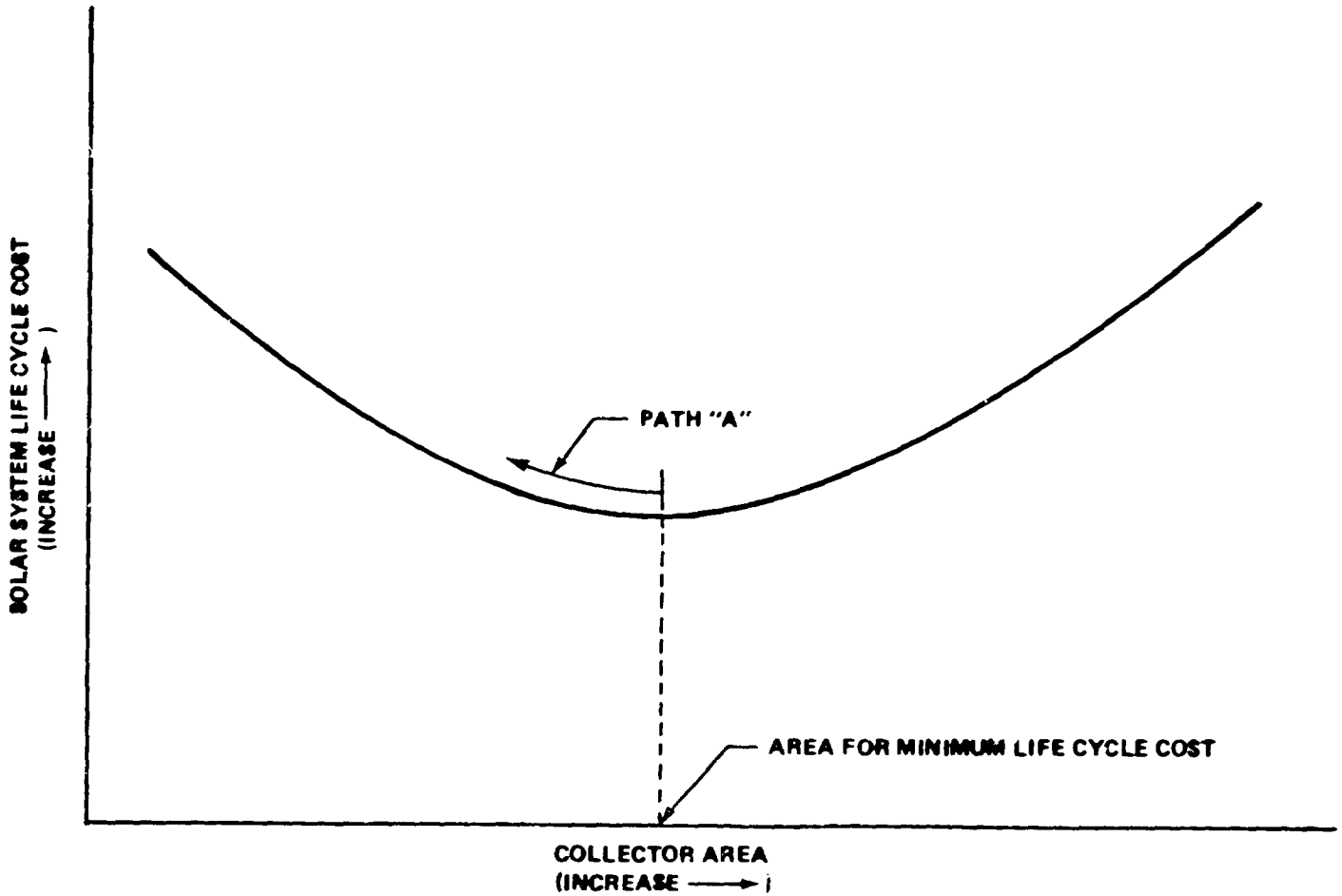


Figure 1. Characteristic curve: life cycle cost versus collector area.

A. Cost Elements

Comparing the life cycle cost of a solar heating system with a conventional system requires, first of all, the identification of costs which are expected to be different over the life of the two alternatives. One of the alternatives is to have a conventional system perform 100 percent of the load using conventional fuel. The second alternative is to have both a solar heating system and a conventional system. In this second alternative, the solar heating system would carry from 50 to typically 70 percent of the total load and the conventional system carry the remaining 30 to 50 percent. The conventional system in this second alternative is sized, however, to have the capability to carry 100 percent of the load if lack of sunlight makes the solar heating system temporarily useless.

The approach adopted in this report is to provide a fairly comprehensive list of cost elements which may be an important part of any economic analysis and let the program user select the ones to include for his particular circumstances. Table 1 lists these cost elements and gives a brief discussion of their meaning. The work of Rosalie Ruegg [3] contains an excellent discussion of methods of economically evaluating solar heating systems. The program described here follows her work closely.

B. Solar Heating System Sizing

Sizing a solar heating system requires knowledge of the total annual heating load of the home. Methods of doing this are readily available [4]. Collector area is considered a variable in this formulation in that the user chooses the one of maximum economic benefit. Starting with an initial collector area and performance characteristics, it is then necessary to determine the percent of the annual load carried by the solar system. There are numerous methods for doing this [2,5]. The GFL Method [5] has been built into the computer program. By selecting a region of the country (from the list of 151 cities in Table 2), a collector area, and the type of working fluid (air or liquid) the computer program determines the percent of total load carried by the solar heating system. The GFL method requires the selection of collector performance parameters. The following values, which are considered typical, are built into the program. For an air system $F'_R\tau\alpha = 0.6$ and $F'_R U_L = 4.0$ W/M^2·°C. For a liquid system $F'_R\tau\alpha = 0.75$ and $F'_R U_L = 4.0$ W/M^2·°C. If the program user has other values for collector performance parameters, they can be input to the program. Also, if the percent solar fraction is known it can be input and the GFL Method will not be used.

TABLE 1. COST ELEMENTS WORKSHEET

Cost Element	Description	Life Cycle Costing Calculation	User Problem Data Worksheet
Acquisition	Initial costs incurred by purchase, delivery, installation and integration.	Treated as initial, one-time cost and is not discounted or inflated. A down-payment factor may be applied.	
Building Modifications	Costs due to structural modifications required for the system.	Treated as initial, one-time cost and is not discounted or inflated.	
Net Replacement and Repair	Yearly cost of replacements and repair to the system.	Cost input in year incurred, then inflated and discounted.	
Maintenance	Estimated annual cost for maintenance of total system.	Initial value input for first year, then inflated and discounted.	
Conventional Fuel Cost	Annual conventional fuel and energy costs required for system operation.	Annual cost for first year input, then inflated by energy escalation factor and discounted.	
Property Taxes	Property taxes paid due to assessed value of the equipment.	Same method as for Maintenance.	
Property Tax Credits	Deductions from income tax due to property taxes paid.	Income tax rate times taxes paid.	
Maintenance Expense Credits	Reduces commercial taxable income.	Income tax rate times expenses.	
Depreciation Credit	Commercial deduction from taxes.	Straight line method, no inflation.	
Added Income	Increased rental of solar compared to conventional property due to lower utilities.	Income is taxed and discounted.	
Insurance	Cost of insurance on the system.	Net annual cost input and discounted, not inflated.	
Salvage	Expected value at end of life.	Discounted.	
Loan Payments	Annual loan payments on borrowed funds.	Loan is amortized and yearly payment computed. Yearly interest computed. Payment is discounted.	
Loan Interest Credits	Tax deduction due to interest paid.	Interest from loan computed and discounted.	
Conventional Fuel Cost Credit	Commercial tax deduction.	Annual cost for first year input, then inflated by energy escalation factor.	

TABLE 2. CITY LISTING

1.	Abilene, Texas	47.	Fort Wayne, Indiana
2.	Albany, New York	48.	Fort Worth, Texas
3.	Albuquerque, New Mexico	49.	Fresno, California
4.	Amarillo, Texas	50.	Gainesville, Florida
5.	Ames, Iowa	51.	Glasgow, Montana
6.	Amherst, Massachusetts	52.	Grand Junction, Colorado
7.	Annapolis, Maryland	53.	Grand Lake, Colorado
8.	Apalachicola, Florida	54.	Great Falls, Montana
9.	Asheville, North Carolina	55.	Green Bay, Wisconsin
10.	Astoria, Oregon	56.	Greensboro, North Carolina
11.	Atlanta, Georgia	57.	Granville-Spartanburg, North Carolina
12.	Atlantic City, New Jersey		
13.	Big Spring, Texas	58.	Griffin, Georgia
14.	Billings, Montana	59.	Hartford, Connecticut
15.	Binghampton, New York	60.	Houston, Texas
16.	Birmingham, Alabama	61.	Indianapolis, Indiana
17.	Bismarck, North Dakota	62.	Inyokern, California
18.	Bluehill, Massachusetts	63.	Ithaca, New York
19.	Boise, Idaho	64.	Jackson, Mississippi
20.	Boston, Massachusetts	65.	Jacksonville, Florida
21.	Boulder, Colorado	66.	Kansas City, Missouri
22.	Brownsville, Texas	67.	Key West, Florida
23.	Cape Hatteras, North Carolina	68.	Lake Charles, Louisiana
		69.	Lander, Wyoming
24.	Caribou, Maine	70.	Lansing, Michigan
25.	Charleston, South Carolina	71.	Laramie, Wyoming
26.	Charlotte, North Carolina	72.	Las Vegas, Nevada
27.	Chattanooga, Tennessee	73.	Lemont, Illinois
28.	Chicago, Illinois	74.	Lexington, Kentucky
29.	Cleveland, Ohio	75.	Lincoln, Nebraska
30.	Columbia, Missouri	76.	Little Rock, Arkansas
31.	Columbus, Ohio	77.	Los Angeles, California
32.	Corpus Christi, Texas	78.	Louisville, Kentucky
33.	Corvallis, Oregon	79.	Lynn, Massachusetts
34.	Dallas, Texas	80.	Macon, Georgia
35.	Davis, California	81.	Madison, Wisconsin
36.	Dayton, Ohio	82.	Manhattan, Kansas
37.	Denver, Colorado	83.	Medford, Oregon
38.	Des Moines, Iowa	84.	Memphis, Tennessee
39.	Detroit, Michigan	85.	Miami, Florida
40.	Dodge City, Kansas	86.	Midland, Texas
41.	Duluth, Montana	87.	Milwaukee, Wisconsin
42.	East Lansing, Michigan	88.	Minn-St. Paul, Minnesota
43.	El Paso, Texas	89.	Mt. Weather, Virginia
44.	Ely, Nevada	90.	Nashville, Tennessee
45.	Fargo, North Dakota	91.	Natick, Massachusetts
46.	Fort Smith, Arkansas	92.	New Orleans, Louisiana

TABLE 2. (Concluded)

93. Newport, Rhode Island
94. New York, New York
95. Norfolk, Virginia
96. North Omaha, Nebraska
97. Oak Ridge, Tennessee
98. Oklahoma City, Oklahoma
99. Page, Arizona
100. Parkersburg, West Virginia
101. Pasadena, California
102. Pensacola, Florida
103. Peoria, Illinois
104. Phoenix, Arizona
105. Philadelphia, Pennsylvania
106. Pittsburgh, Pennsylvania
107. Pocatello, Idaho
108. Port Arthur, Texas
109. Portland, Maine
110. Portland, Oregon
111. Prosser, Washington
112. Pueblo, Colorado
113. Pullman, Washington
114. Put-in-Bay, Ohio
115. Raleigh, North Carolina
116. Raleigh-Durham, North Carolina
117. Rapid City, South Dakota
118. Reno, Nevada
119. Richland, Washington
120. Richmond, Virginia
121. Riverside, California
122. Rochester, New York
123. Sacramento, California
124. St. Cloud, Minnesota
125. St. Louis, Missouri
126. Salt Lake City, Utah
127. San Antonio, Texas
128. San Diego, California
129. San Francisco, California
130. San Jose, California
131. Santa Maria, California
132. Savannah, Georgia
133. Sault St. Marie, Michigan
134. Schenectady, New York
135. Seattle, Washington
136. Shreveport, Louisiana
137. Silver Hill, Maryland
138. Spokane, Washington
139. State College, Pennsylvania
140. Stillwater, Oklahoma
141. Summit, Montana
142. Syracuse, New York
143. Tallahassee, Florida
144. Tampa, Florida
145. Trenton, New Jersey
146. Tucson, Arizona
147. Tulsa, Oklahoma
148. Twin Falls, Idaho
149. Washington, D.C.
150. Wichita, Kansas
151. Yuma, Arizona

C. Sensitivity Studies

Selecting values of certain key cost parameters, such as future conventional fuel escalation rates, requires subjective judgment. The program is designed to study a range of values of certain parameters in one run or execution of the program. The parameters which can be varied are collector area, discount rate, inflation rate, down payment factor, property tax rate and income tax rate. The plot feature of the program produces plots of life cycle cost as a function of any of these parameters.

II. USER PROBLEM FORMULATION

The total life cycle cost of the solar and conventional energy system is computed by adding the individual life cycle costs of the basic cost elements which are appropriate for the type of analysis desired. The user selects the basic cost elements which he considers important. There are fifteen possible cost elements (Table 1) covering an extensive range of possibilities for residential and commercial applications.

The life cycle cost calculation is formulated for computer solution in the classic Work Breakdown Structure (WBS) format. See Reference 6 for a description of the WBS concept. In this application of the WBS concept each cost element is a block in the WBS where costs are accumulated from sub-elements at a lower level and are summed to cost elements at a higher level. A three level WBS is illustrated in Figure 2. The highest level block, Solar System Life Cycle Cost, is a Level I block and contains the total life cycle costs of the solar system. There are 7 Level II blocks illustrated from Acquisition to Loan Interest Credits. Note that the Level II block titles are cost elements from Table 1. Level III is illustrated by showing that Acquisition has been further sub-divided into various hardware components as well as Integration and Installation.

The first step in formulating the user problem is to select from Table 1 the cost elements for the solar and conventional energy system to be analyzed. It may save some work for the user to recall that when comparing life cycle costs for two competing systems, it is only necessary to consider cost elements which may be different for the two systems. For example, if no significant difference in property tax is expected between the solar and conventional system, then it is not necessary to include that cost element in the analysis of either.

The next step is to construct a WBS. This is a very important step and must be done properly for successful program execution. The key decisions which the user must make for each WBS block are its proper level, title, WBS number and block number. Figure 2 illustrates these concepts.

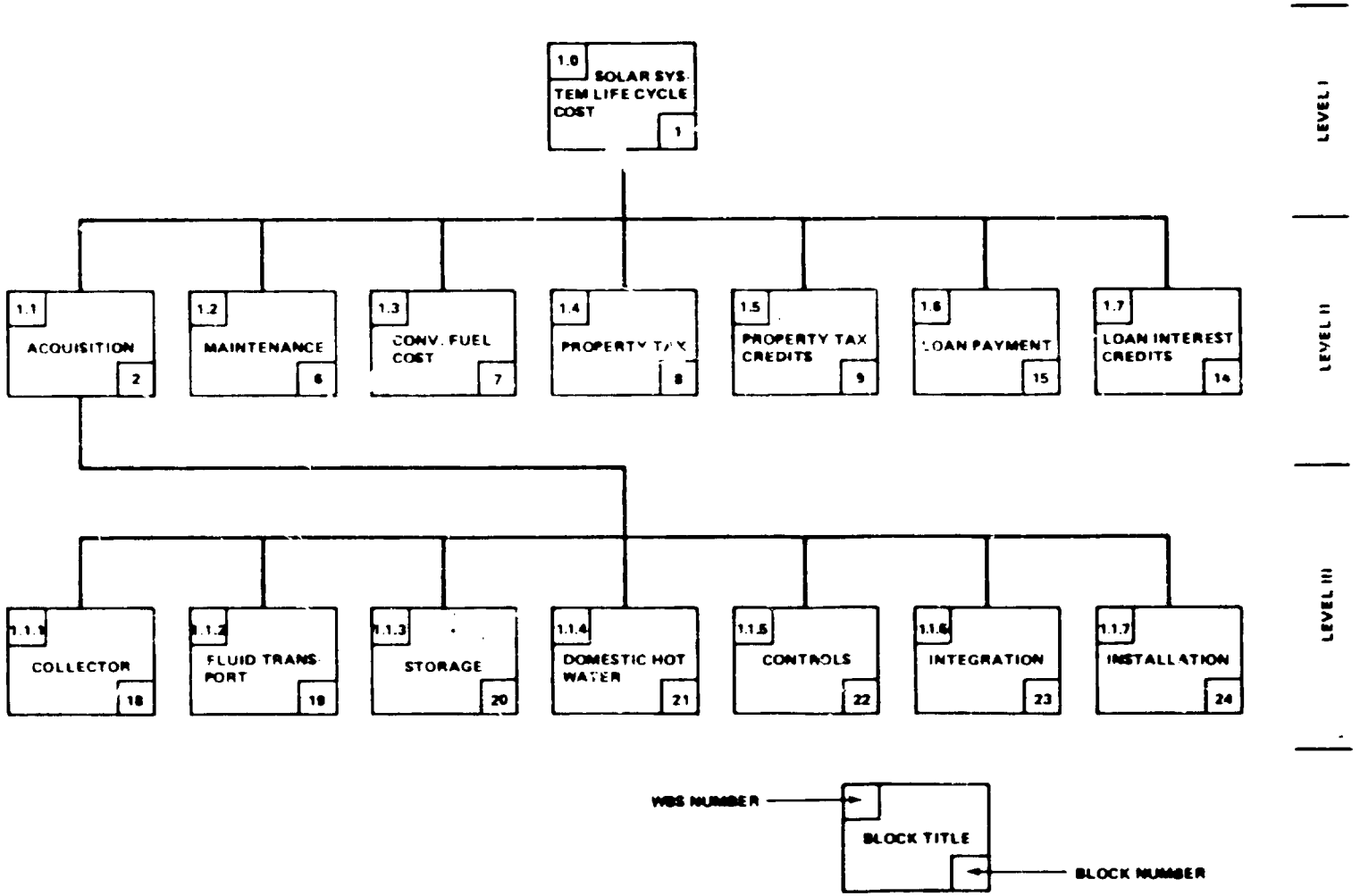
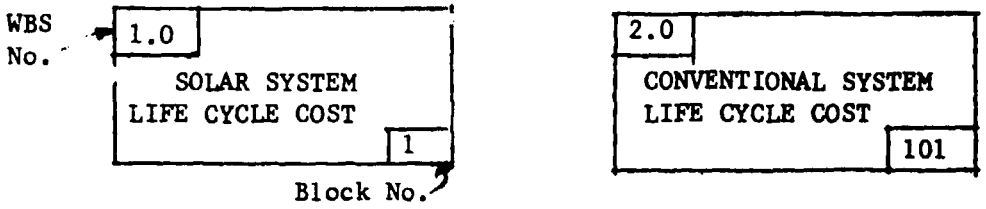


Figure 2. WBS for typical solar system life cycle cost analysis.

A. Rules for Constructing the WBS

1. Level I Blocks. Two (no more, no less) Level I blocks are necessary. They must have the following titles, WBS numbers and block numbers:



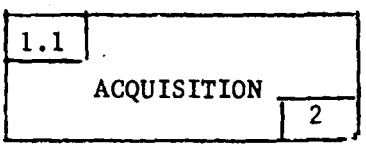
There will usually be several Level II blocks under each of these Level I blocks.

2. Level II Blocks.

a) Each Level II block must be one of the cost elements from Table 1.

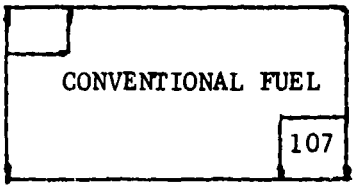
b) There will be two groups of Level II blocks, one group under the Level I block Solar System Life Cycle Cost, and the other under the Level I block Conventional System Life Cycle Cost.

c) The first Level II block under WBS 1.0 must be:



This block must be included in each problem and have the indicated WBS number, title and block number. There are usually several other Level II blocks under WBS 1.0.

d) The other Level I block, WBS 2.0, must contain the following Level II block with indicated title and block number:



Note that no specific WBS number is required. A logical number to assign to it will become evident to the user as the construction of the WBS continues.

e) The number of additional Level II blocks under each Level I block depends on which additional cost elements the user decides to consider.

f) When a cost element is selected and made a Level II block, its title and block number are restricted. Block titles at Level II are always the same as cost elements from Table 1. Block numbers at Level II are restricted to the values shown in Table 3.

g) Remaining to be selected are the WBS numbers for Level II blocks. From an understanding of the WBS concept, logical numbers under the Level I block number 1.0 would be 1.1, 1.2, 1.3, etc. and under 2.0 would be 2.1, 2.2, 2.3, etc.

TABLE 3. COST ELEMENT BLOCK NUMBERS

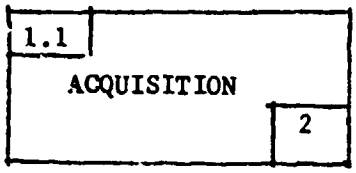
Block Title (Cost Element)	When Selected to Go Under 1.0, Must Have the Block Number	When Selected to Go Under 2.0, Must Have the Block Number
Acquisition	2	102
Building modifications	3	103
Space occupied	4	104
Repair	5	105
Maintenance	6	106
Conventional fuel	7	107
Property taxes	8	108
Property tax credits	9	109
Depreciation credits	10	110
Added income	11	Not applicable
Insurance	12	111
Salvage	13	112
Loan payments	15	114
Loan interest credits	14	113
Maintenance expense credits	16	115
Fuel cost credits	17	116

3. Level III Blocks.

a) The purpose of having Level III blocks available is to allow the user to specify additional detail, if desired, for each Level II block he selects. For example, under the cost element salvage value for a solar system, the user may choose to break down the total salvage value due to the collectors, hot water system, controls, etc.

b) There is one level III block required in all problems,

and it must have the indicated WBS number, block title and block number. It must go under the Level II block:



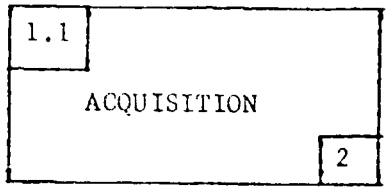
c) When Level III blocks are used, the user is in effect creating sub-cost elements under a Level II block. The sub-cost elements are treated the same mathematically as the Level II cost element they are under. It is important that the sum of all the Level III block costs under a particular Level II block equal what would have been input for the Level II block cost had the user not elected to go to the Level III detailed breakout. Additionally, after having elected to go to Level III detail, it is not necessary to input the data at Level II since the computer will sum the Level III input data and results at Level II.

d) WBS numbers and block titles at level III can be selected by the user (except for as previously noted).

e) Block numbers at Level III are restricted to the following values:

Level III blocks under the Level I block 1.0 can have values between 19 and 100. Level III blocks under the Level I block 2.0 can have values between 118 and 200.

f) Note that forcing the user to go to Level III under the Level II block



requires the creation of additional Level III blocks under 1.1, besides 1.1.1 Collector to handle acquisition costs of other components in the solar system as well as integration and installation costs. See Figure 3 as an illustration. The number of these additional Level III blocks is at the discretion of the user.

Having now been through the rules for constructing the WBS, the WBS for the user's problem should be properly constructed. Each WBS block should have a unique WBS number and block number. The WBS numbers should now be arranged in the classic WBS indentured format:

1.0 Solar System Life Cycle Cost
- 1.1 Acquisition
  - 1.1.1 Collector
  - 1.1.2 Other
- 1.2 Building Modifications
- •
- •
- •

2.0 Conventional System Life Cycle Cost
- 2.1 Operations
- •
- •
- •

A diagram similar to Figure 3 showing the WBS blocks at each level and their relation to each other should be constructed for the user problem.

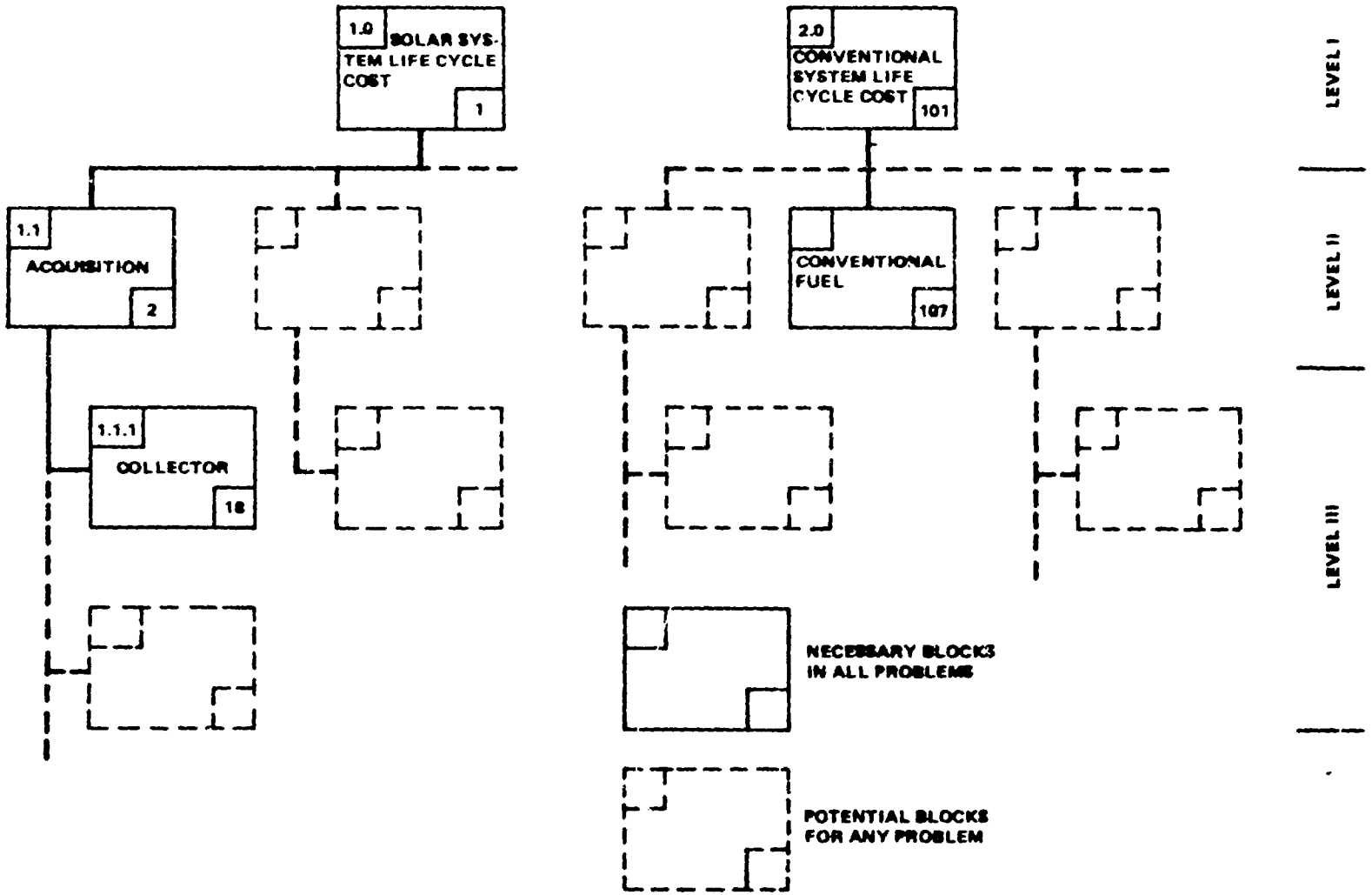


Figure 3. Additional Level III block possibilities.

B. Data Collection

The next step in formulating the user problem is to gather the necessary cost data and related problem parameters. Data for each selected cost element from Table 1 is required. The additional cost data and problem parameters needed for formulating the complete problem are outlined in Tables 4 and 5. Table 6 indicates the units and format required for all data.

TABLE 4. ADDITIONAL COST DATA WORKSHEET

Additional Cost Data	User Problem Worksheet
Collector cost in dollars/m^2	
Discount Rate	
Inflation Rate	
Down Payment Rate	
Property Tax Rate	
Income Tax Rate	

TABLE 5. PROBLEM PARAMETER WORKSHEET

Problem Parameters	User Problem Worksheet
Number of years in the life cycle cost analysis	
Starting year	
Collector area, m^2	
City location code number	
Liquid or Air system	
Total building load, GJ/yr	
Commercial or residential application	

TABLE 6. DATA UNITS AND COMPUTER FORMAT

Cost Data/Problem Parameter	Units	Computer Input Format
Collector Cost	$\$/m^2$	F10.0
Total Building Load	GJ/Yr	F10.0
Discount Rate	Percent (10 percent input as 0.1)	F8.2
Inflation Rate	Percent (6 percent input as 0.06)	F8.2
Downpayment Factor	Percent (10 percent input as 0.1)	F8.2
Property Tax Rate	Percent (2 percent input as 0.02)	F8.2
Income Tax Rate	Percent (30 percent input as 0.3)	F8.2
Collector Area	m^2	F8.2
Mortgage Interest Rate	Percent (9 percent input as 0.09)	F8.2

III. USER PROBLEM SOLUTION

The solution is obtained by inputing the user problem data into the computer in proper format and then executing the program. To begin the process of inputing the problem data, the user should have on hand the following information:

1) The WBS in block diagram form (Fig. 4) with appropriate WBS numbers, block titles and block numbers.

2) The WBS in indentured format (Table 7).

3) Cost data for the selected cost elements from Table 1. The blank column in Table 1 is provided so that it can also be used as a problem worksheet.

4) The additional cost data and problem parameters as outlined in Tables 4 and 5. The blank column in Tables 4 and 5 is provided so that it can also be used as a problem worksheet.

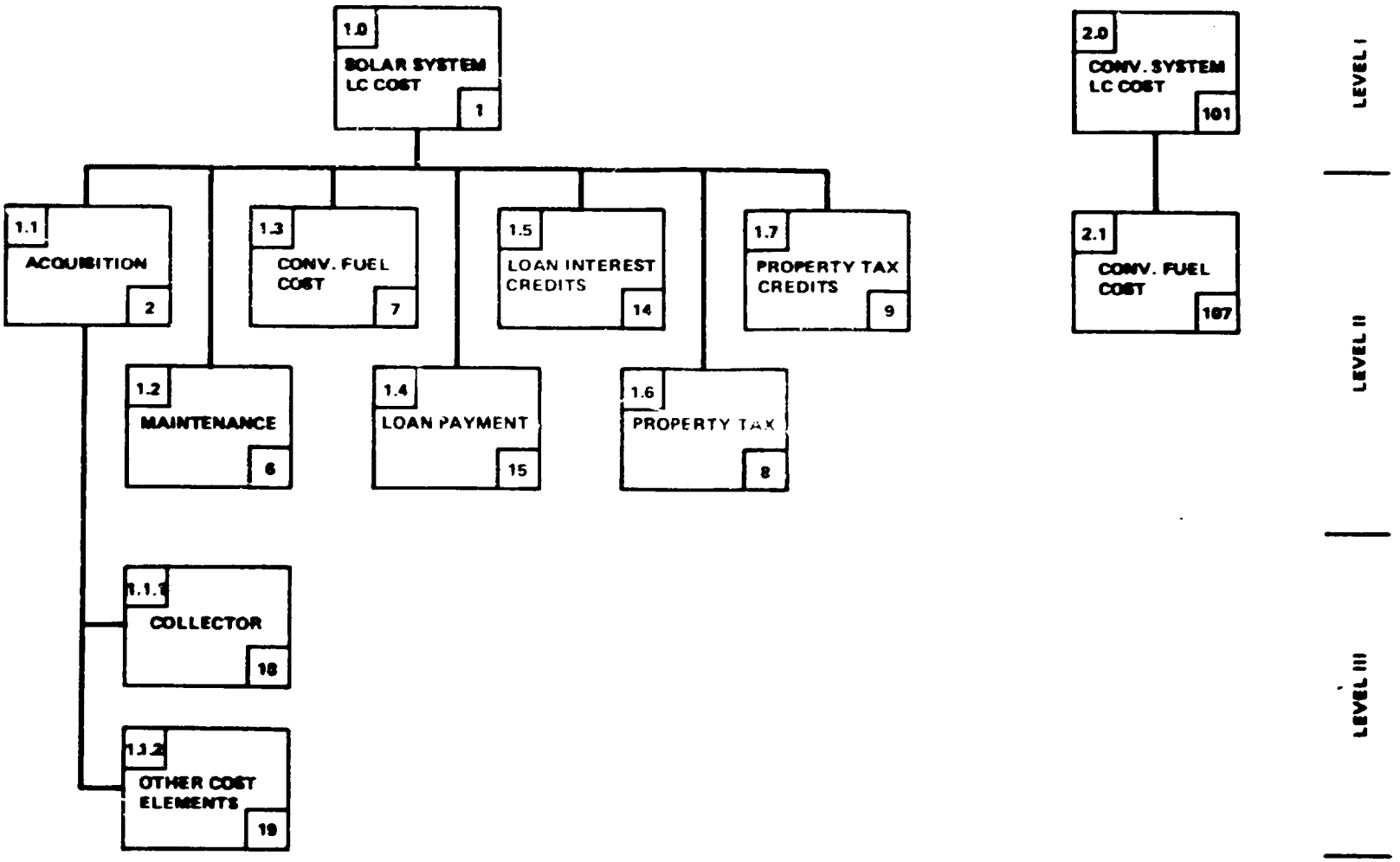


Figure 4. Sample problem WBS structure.

TABLE 7. SAMPLE PROBLEM WBS INDENTURED FORMAT LISTING

1.0 Solar System Life Cycle Cost
1.1 Acquisition
1.1.1 Collector
1.1.2 Other Cost Elements
1.2 Maintenance
1.3 Conventional Fuel Cost
1.4 Loan Payments
1.5 Loan Interest Credits
1.6 Property Tax
1.7 Property Tax Credits
2.0 Conventional System Life Cycle Cost
2.1 Conventional Fuel Cost

All the problem data will now be transferred to user input sheets. User input sheets are in standard 80 column punched card format and are illustrated in Figures 5, 6 and 7. The user input sheets are divided into three sections and the input rules for each will be described. Generally speaking, Section I input defines the WBS structure of the user problem and establishes which cost elements are to be considered in the life cycle cost analysis. Section II defines certain parameters and data values of the user problem. The main data input for each cost element occurs in Section III. Units and computer input format for all data are described in Table 6. The first card in the deck, after the run card, is a title card. Any appropriate descriptive title for the user problem up to 30 characters in length may be used. See Figure 8 for an example of how a complete deck is assembled.

A. Preparation of Input Data

1. Section I. Section I is illustrated in Figure 5. The first step is to write in the WBS-numbers (in Columns 4-8) and the block titles (in Columns 9-32) using the indentured WBS listing of the user problem as a guide to the proper sequence. There should be one card in Section I for each block in the WBS and they should be in the sequence of the indentured format listing.

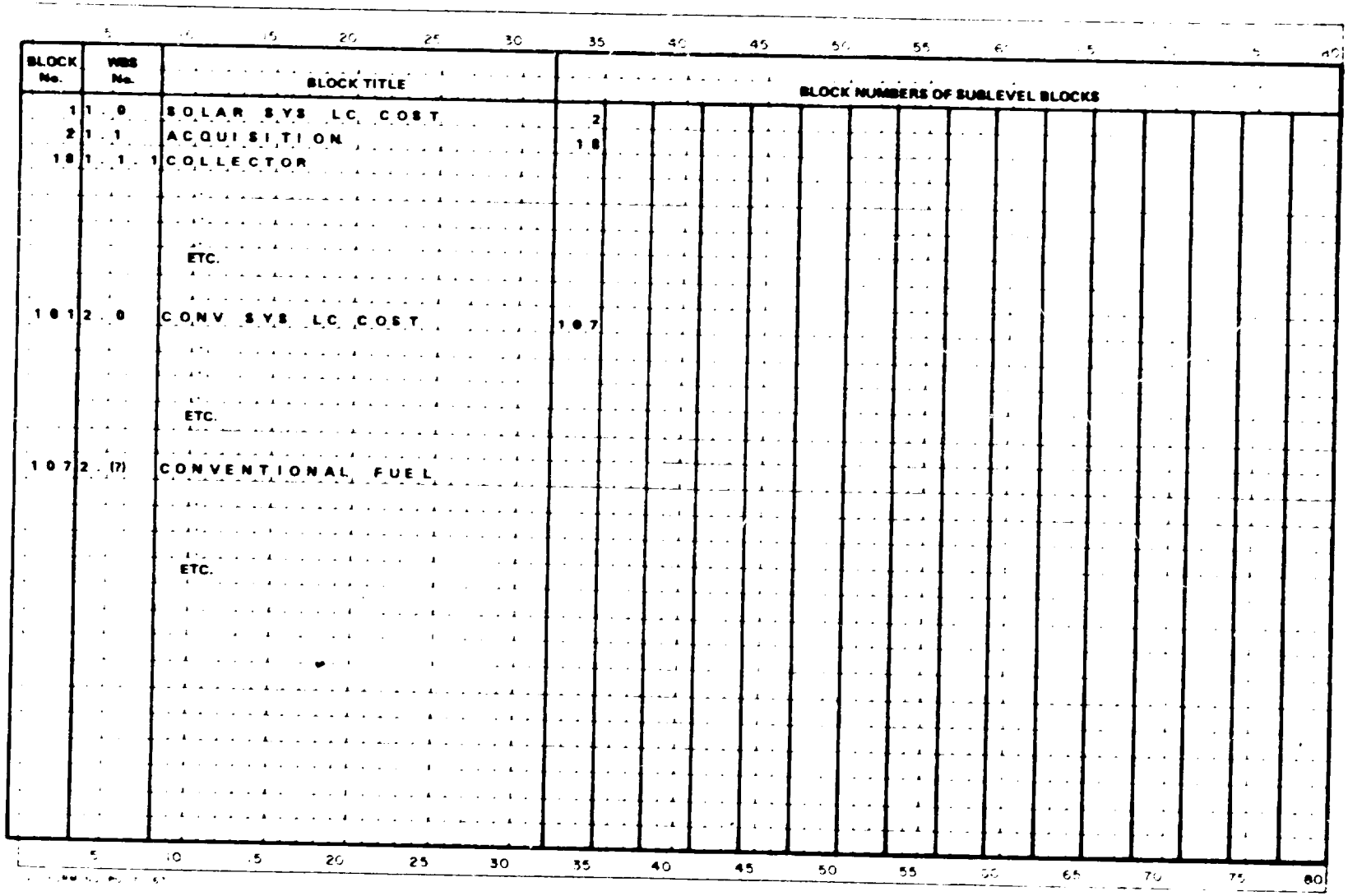


Figure 5. User problem input sheet - Section 1.

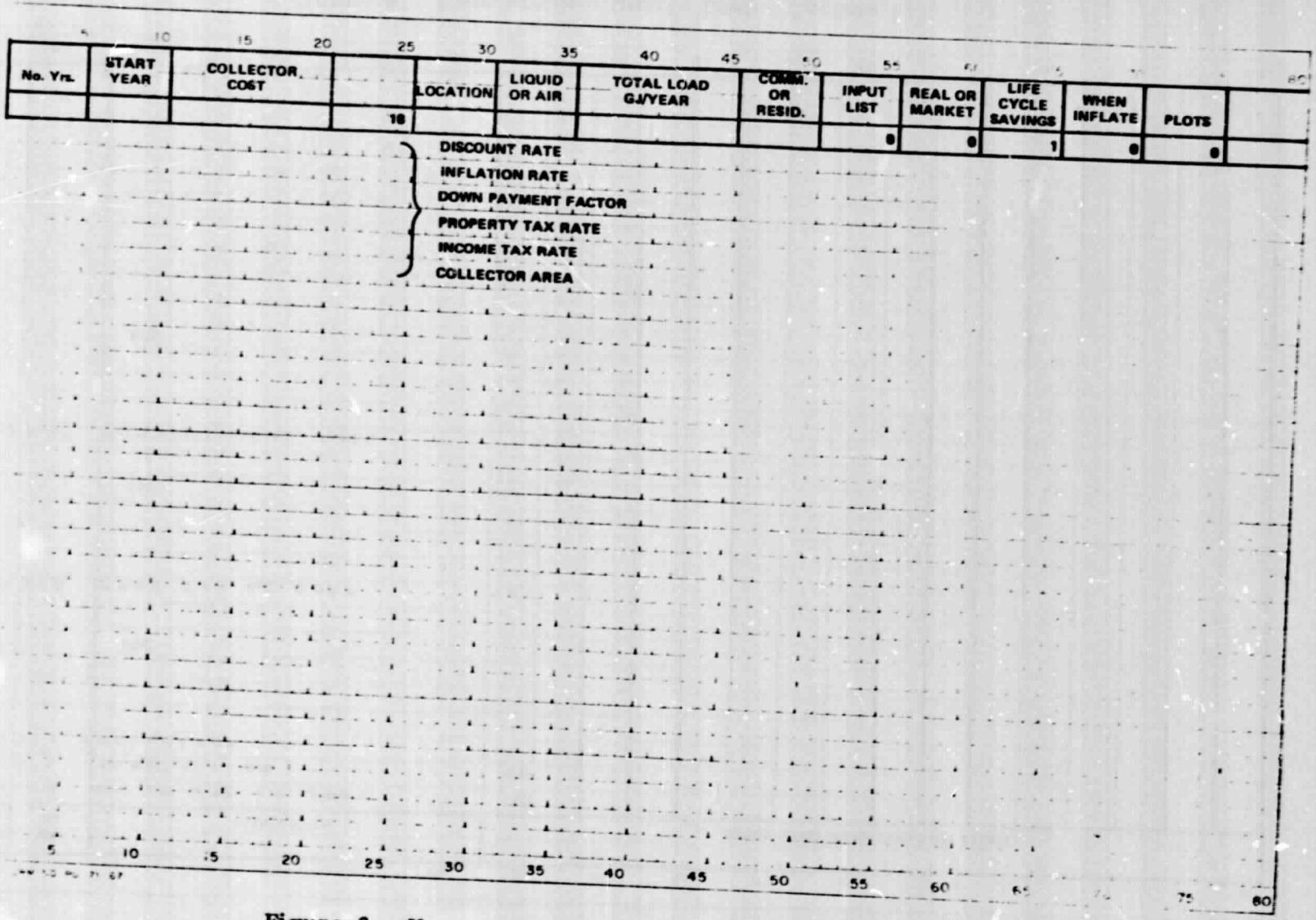


Figure 6. User problem input sheet - Section II.

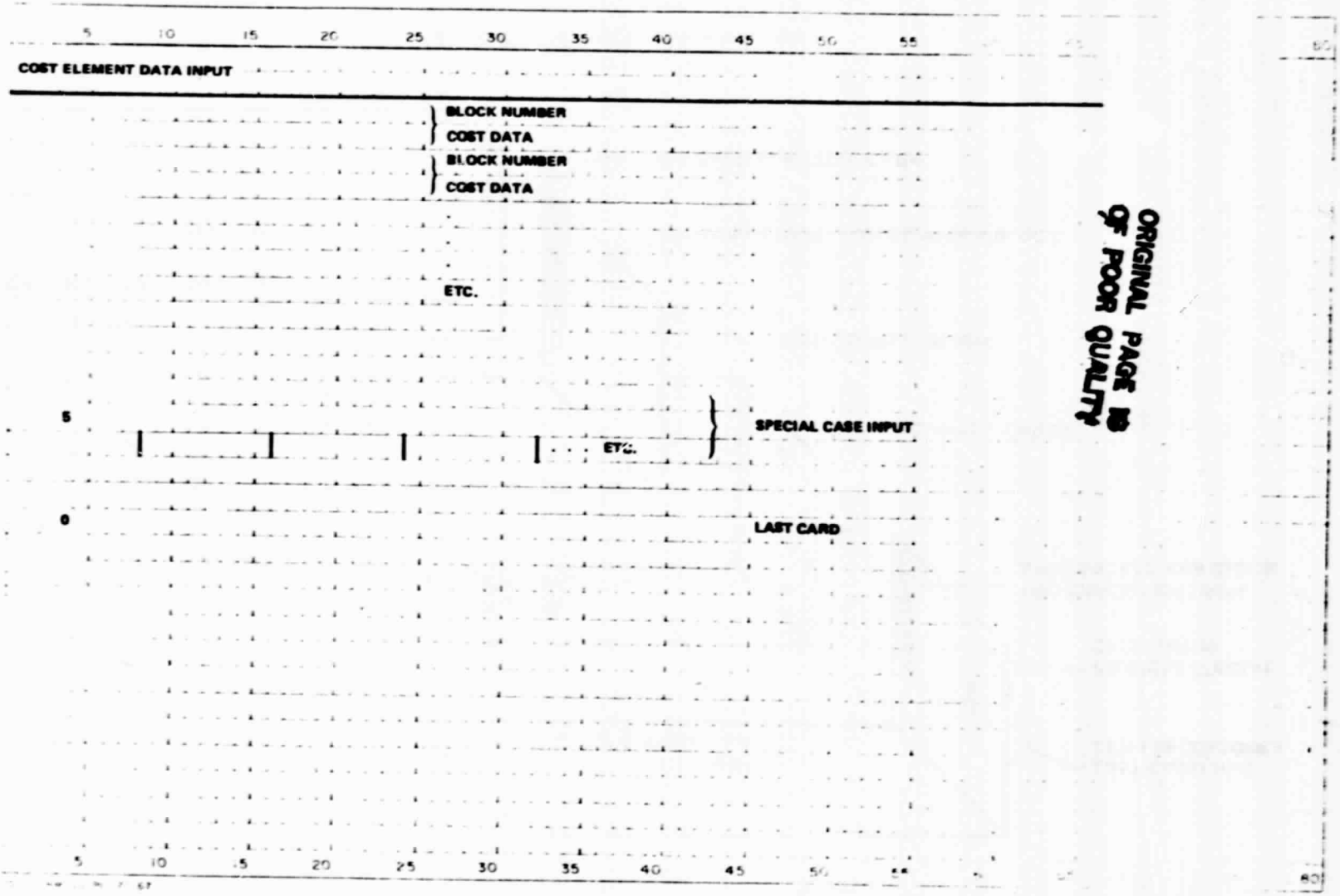


Figure 7. User problem input sheet - Section III.

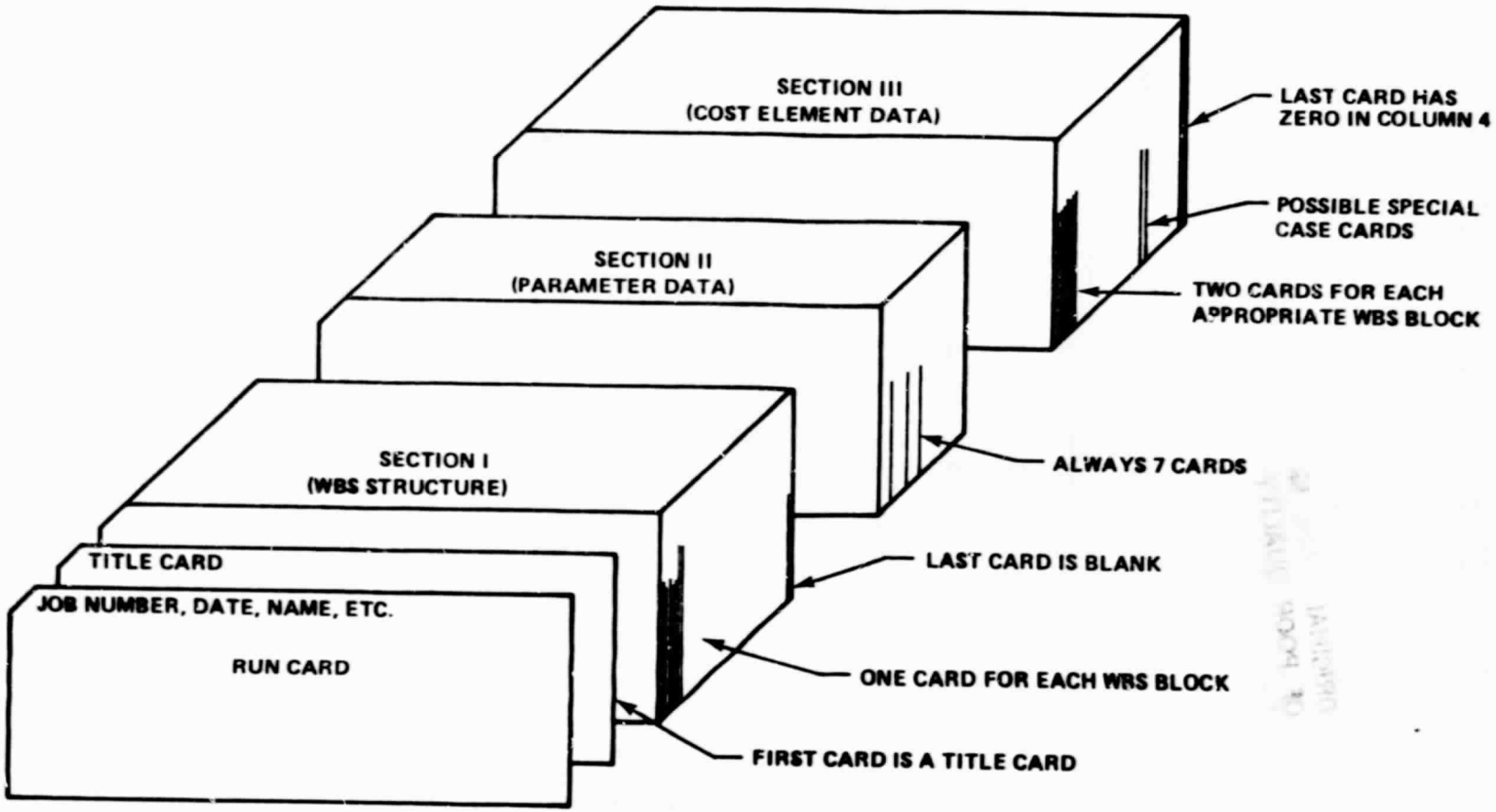


Figure 8. Typical deck setup.

The next step is to write in the block numbers (in Columns 1-3) using the WBS block diagram to obtain block numbers. Note that in Figure 5 five blocks are illustrated. It should be recalled from the discussion of rules for constructing the WBS that these are the 5 blocks that must be a part of every problem. The first three (1.0, 1.1 and 1.1.1) are in the sequence they will always be in for any problem. The proper location of 2.0 and 2.(?) in the sequence will depend on the number of other blocks in the indentured format listing of the user problem.

It is now necessary to insert the block numbers of sublevel blocks in the field of Columns 33-80 for each WBS block. The rules for completing this portion of the input are described in the following paragraphs. The objective of this input is to tell the computer which blocks have lower level blocks under them and for those that do, what the sublevel block numbers are.

Rules for Section I input of block numbers of sublevel blocks are presented in the following paragraphs.

a. Level III Blocks –

1) Since Level III blocks are the lowest in the WBS they have no subblocks. Therefore, no input is appropriate. Leave Columns 33-80 blank on the card for each Level III block in the indentured listing.

b. Level II Blocks –

1) For the Level II blocks which have no Level III blocks under them, simply leave the field blank.

2) For Level II blocks with Level III subblocks, insert each Level III subblock number in the field of Columns 33-80. They do not have to be in any specific sequence. Note that one of the Level III block numbers goes in Columns 33-35, another block number in Columns 36-38, another in 39-41, etc., until all Level III block numbers under that Level II block are listed. Note as an example in Figure 5 that Block Number 18 (a Level III block) is listed as being a sublevel block of Block Number 2, 1.1, Acquisition, itself a Level II block.

c. Level I Blocks –

1) Each of the two Level I blocks (WBS No. 1.0 and 2.0) must have at least one Level II block under it according to our previous discussions and usually there are several other Level II blocks.

2) Insert the Level II block numbers under WBS 1.0 on the card for WBS 1.0 and in the field 33-80. Note that in Figure 5, Block Number 2 is listed.

3) Do not put Level III block numbers on the card of a Level I block, although, in a sense, Level III blocks are sublevels of a Level I block.

4) Insert the Level II block numbers under WBS 2.0 on the card of WBS 2.0 and in the field 33-80. Note in Figure 5 that Block Number 107 is already listed.

There should be as many cards in Section I as there are WBS blocks in the user problem. A blank card should be added as the last card. The input for this section is now complete.

2. Section II. Data for inputing this section comes from Tables 4 and 5. Figure 6 shows how the various data are input. The first card contains the following parameters and data:

a) The number of years in the life cycle cost analysis goes in Columns 1-5.

b) The starting year of the analysis goes in Columns 6-10.

c) The collector cost in $\$/m^2$ goes in Columns 11-20.

d) Columns 24 and 25 must have "18" in them.

e) The city location code from Table 2 goes in Columns 26-30.

f) For a liquid system insert 1 in Column 35. For an air system insert 2 in Column 35.

g) The total building load in GJ/yr goes in Columns 36-45.

h) For a residential problem input 0 in Column 50. For a commercial problem input 1 in Column 50.

i) The remaining Columns (51-80) should contain the numbers indicated in Figure 3.

The remaining six cards in Section II contain the following data, all inputs beginning in Column 1: discount rate, inflation rate, down payment factor, property tax rate, income tax rate, and collector area. This completes input for Section II.

3. Section III. Figure 7 illustrates the format for Section III input. The purpose of Section III is to input the user problem cost and parameter data for certain WBS blocks. Not all WBS blocks require an input in this section. The WBS blocks of the user problem which require input in this section are identified by the following rules:

a) Neither of the Level I blocks require data input.

b) The following Level II blocks, if they happen to be a part of the user problem, do not require input in this section: Block Number 7, Conventional Fuel Cost; Block Number 8, Property Taxes; Block Number 9, Property Tax Credits; Block Number 15, Loan Payments; Block Numbers 16 and 115, Maintenance Expense Credits; Block Numbers 17 and 116, Conventional Fuel Cost Credits.

c) Every other Level II block requires data input unless it has Level III subblocks.

d) All Level III blocks require data input except Block Number 18.

Two cards are required to complete the input for each block. The first card contains the block number and the second contains the cost/parameter data. The block card pairs can be grouped in any order; the important point is that the cost-parameter data for each block be on the card immediately following the card containing the block number. The WBS together with these rules should identify the WBS blocks which require cost/parameter data input in Section III. A description of how to input the cost/parameter data on the second card of the block card pair is provided in Table 8. The last portion of the Section III input is for the special case when net replacement and repair has been selected as a cost element in the user problem. In this case, at least two additional cards are required. The first must contain a "5" in Column 4 as illustrated in Figure 7. On the second card, the cost incurred in the first year of the analysis is input in Columns 1 to 8, the cost incurred in the second year of the analysis is input in Columns 9 to 16, those for the third year in 17 to 24, etc. More than this second card may be required to input the data for all years of the analysis, depending on the number of years in the life cycle cost analysis. The very last card of Section III must contain a zero in Column 4.

B. Program Execution

The program is designed to run on the Univac 1108 with EXEC 8 operating system. The core size is 32k. The program is written in ASC II Fortran and runs in batch mode. A Cal-Comp plotter is used for output when a sensitivity run is made and plots are required. Figure 8 shows a typical deck setup with notes that are consistent with the previously discussed rules for inputing the data. Section VI of this report, Program Documentation, contains additional information on program execution.

TABLE 8. SECTION III COST/PARAMETER DATA INPUT

Cost Element	Cost/Parameter Data	Input Description
Building Modifications	Cost in Dollars	Begins in Column 1, F8.2
Acquisition	Cost in Dollars	Begins in Column 1, F8.2
Maintenance	Percent of Acquisition Cost in First Year	Begins in Column 1, F8.2
Conventional Fuel Cost	Cost in Dollars for Supplying 100 Percent of Load and Escalation Rate in Percent	Cost Beginning in Column 1 (F8.2) Followed by Escalation Rate (F8.2)
Depreciation Credit	Salvage Value in Dollars After N Years of Depreciation	Salvage Value Beginning in Column 1 (F8.2) Followed by N (F8.2)
Added Income	Income in Dollars for First Year	Begins in Column 1, F8.2
Insurance	Percent of Acquisition Cost in First Year	Begins in Column 1, F8.2 0. Should be in Columns 10 and 11 and also in Columns 20 and 21.
Salvage	Value in Dollars at End of Period of Analysis	Begins in Column 1, F8.2
Load Interest Credits	Number of Years of Mortgage and Percent Interest Rate	Number of Years Begins in Column 1, (F8.2), Followed by Interest Rate (F8.2).

IV. SAMPLE PROBLEM

As an example of the formulation and solution of a problem, a sample case is described. This particular problem has been used by the Systems Simulation and Economic Analysis Working Group,[1] a DOE sponsored organization, to compare economic analysis programs nationwide. The sample problem is defined in Table 9. As previously recommended, Tables 1, 4 and 5 have been used as problem worksheets and appear here as Tables 10, 11 and 12. The WBS for the problem is shown in Figure 4 and the WBS indentured format listing is in Table 7.

Since no collector area or geographic location is specified, it is necessary to select some values for purposes of executing the program. The $10,000 initial system cost has been arbitrarily split into $8,000 for collectors and $2,000 for other cost elements. In Birmingham, Alabama, a collector area of 48.28 m^2 is necessary to carry 70 percent of the 100 MBtu/yr load. All the information is now available to prepare the computer input sheets. The results of that are shown in Figures 9, 10 and 11. The program was then executed and the problem results are shown in Table 13 and Figure 12.

TABLE 9. SAMPLE PROBLEM DATA

Initial Cost of System	$10,000
Downpayment	10 percent
Mortgage Rate	9 percent
Discount Rate	10 percent
Maintenance (percent of Initial Cost)	1 percent/yr
Property Tax (percent of Initial Cost)	2 percent
General Inflation Rate	6 percent
Fuel Inflation Rate	10 percent
Effective Tax Rate	30 percent
Initial Fuel Cost	10 $/MBtu
Load Demand	100 MBtu/yr
Percentage Solar	70 percent
System and Mortgage Lifetime	20 yr
Salvage Value	0

1. "System and Economic Analysis Meeting Report," J. M. Alcone, Sandia Laboratories, Albuquerque, New Mexico, March 4, 1977.

TABLE 10. SAMPLE COST ELEMENTS WORKSHEET

Cost Element	Description	Life Cycle Costing Calculation	User Problem Data Worksheet
Acquisition	Initial costs incurred by purchase, delivery, installation and integration.	Treated as initial, one-time cost and is not discounted or inflated. A down-payment factor may be applied.	$10,000 total, arbitrarily split into $8,000 collector and $2,000 other cost elements.
Building Modifications	Costs due to structural modifications required for the system.	Treated as initial, one-time cost and is not discounted or inflated.	NA
Net Replacement and Repair	Yearly cost of replacements and repair to the system.	Cost input in year incurred, then inflated and discounted.	NA
Maintenance	Estimated annual cost for maintenance of total system.	Initial value input for first year, then inflated and discounted.	1 percent of initial total cost/yr i.e. $100/yr.
Conventional Fuel Cost	Annual conventional fuel and energy costs required for system operation.	Annual cost for first year input, then inflated by energy escalation factor and discounted.	SOLAR: $300 first year for conventional fuel. CONVENTIONAL: $1,000 first year
Property Taxes	Property taxes paid due to assessed value of the equipment.	Same method as for Maintenance.	2 percent of initial cost.
Property Tax Credits	Deductions from income tax due to property taxes paid.	Income tax rate times taxes paid.	To be considered.
Maintenance Expense Credits	Reduces commercial taxable income.	Income tax rate times expenses.	NA
Depreciation Credit	Commercial deduction from taxes.	Straight line method, no inflation.	NA
Added Income	Increased rental of solar compared to conventional property due to lower utilities.	Income is taxed and discounted.	NA
Insurance	Cost of insurance on the system.	Net annual cost input and discounted, not inflated.	NA
Salvage	Expected value at end of life.	Discounted.	0, therefore not considered.
Loan Payments	Annual loan payments on borrowed funds.	Loan is amortized and yearly payment computed. Yearly interest computed. Payment is discounted.	$9,000 borrowed at 9 percent for 20 years.
Loan Interest Credits	Tax deduction due to interest paid.	Interest from loan computed and discounted.	To be considered.
Conventional Fuel Cost Credit	Commercial tax deduction.	Annual cost for first year input, then inflated by energy escalation factor.	NA

TABLE 11. COST DATA WORKSHEET

Additional Cost Data	User Problem Worksheet
Collector cost in dollars/m^2	\$8,000/48.28 m^2 = \$165.70/m^2
Discount Rate	10 percent
Inflation Rate	6 percent
Down Payment Rate	10 percent
Property Tax Rate	2 percent of Initial Cost
Income Tax Rate	30 percent

TABLE 12. PROBLEM PARAMETERS SAMPLE WORKSHEET

Problem Parameters	User Problem Worksheet
Number of years in the life cycle cost analysis	20 years
Starting year	1980
Collector area, m^2	48.28 m^2
City location code number	16, Birmingham, Alabama
Liquid or Air system	Liquid
Total building load, GJ/yr	100 MBtu/yr = 105.6 GJ/yr
Commercial or Residential application	Residential

For this sample problem, it is cost effective to install a solar heating system. A life cycle cost savings of \$2,915.18 is achieved. A cash flow diagram of actual yearly costs is shown in Figure 12. Positive savings are first realized in 1983 and "payback" occurs in 1987. The computer printout for the sample problem is also presented in Appendix A. Superimposed on the computer printout are notes explaining various output values.

V. PROGRAM FLEXIBILITY

The previous discussion of how to formulate and solve a user problem, together with the sample problem described, is intended to develop in the user a basic understanding of how to use the program. The program has been designed with the flexibility to solve a wide range of specific problems in evaluating economic feasibility. In this section those features of the program are described.

BLOCK No.	WRS No.	BLOCK TITLE	BLOCK NUMBERS OF SUBLEVEL BLOCKS																
1	1.0	SOLAR SYSTEM LC COST	2	6	7	15	14	8	9										
2	1.1	ACQUISITION	18	19															
18	1.1.1	COLLECTOR																	
19	1.1.2	OTHER COST ELEMENTS																	
6	1.2	MAINTENANCE																	
7	1.3	CONV FUEL COST																	
15	1.4	LOAN PAYMENTS																	
14	1.5	LOAN INTEREST CREDITS																	
8	1.6	PROPERTY TAXES																	
9	1.7	PROPERTY TAX CREDITS																	
101	2.0	CONV SYSTEM LC COST	107																
107	2.1	CONV FUEL COST																	

Figure 9. User problem input sheet - Section I sample problem.

No. Yrs.	START YEAR	COLLECTOR COST		LOCATION	LIQUID OR AIR	TOTAL LOAD GJ/YEAR	COMM. OR RESID.	INPUT LIST	REAL OR MARKET	LIFE CYCLE SAVINGS	WHEN INFLATE	PLOTS	
20	1980	165.70	18	16	1	105.6	0	0	0	1	0	0	

0.1	DISCOUNT RATE
0.06	INFLATION RATE
0.1	DOWN PAYMENT FACTOR
0.02	PROPERTY TAX RATE
0.3	INCOME TAX RATE
48.28	COLLECTOR AREA

Figure 10. User problem input sheet - Section II sample problem.

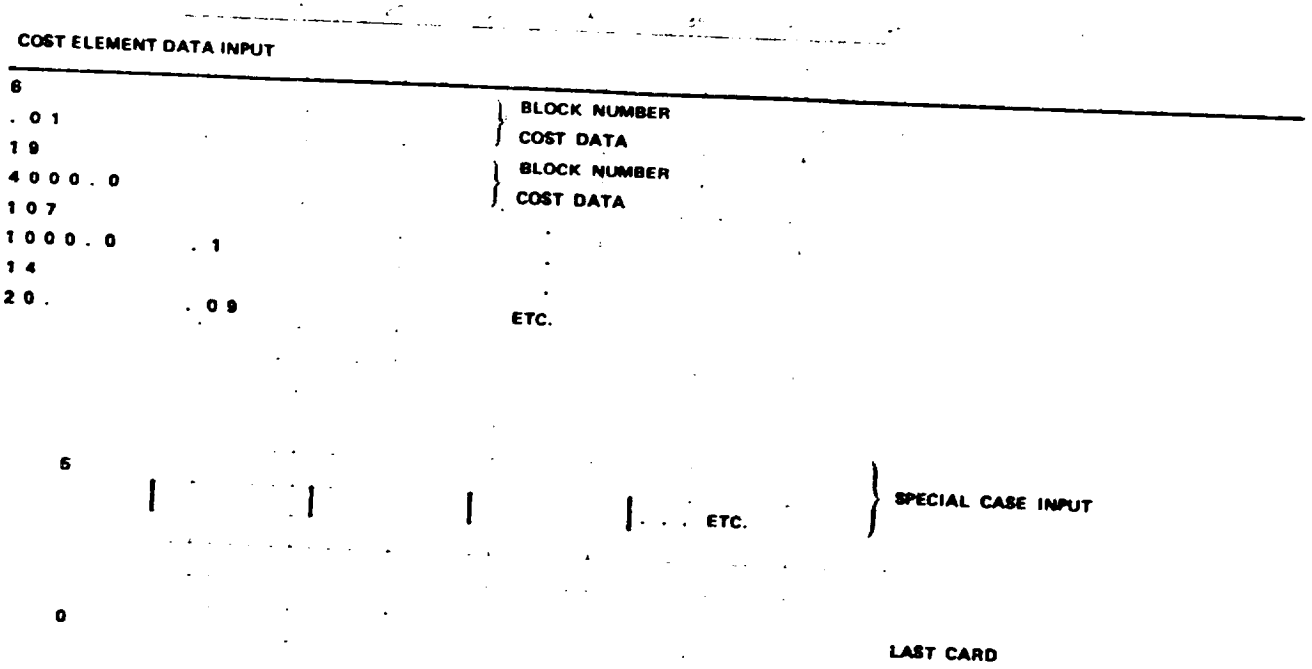


Figure 11. User problem input sheet - Section III sample problem.

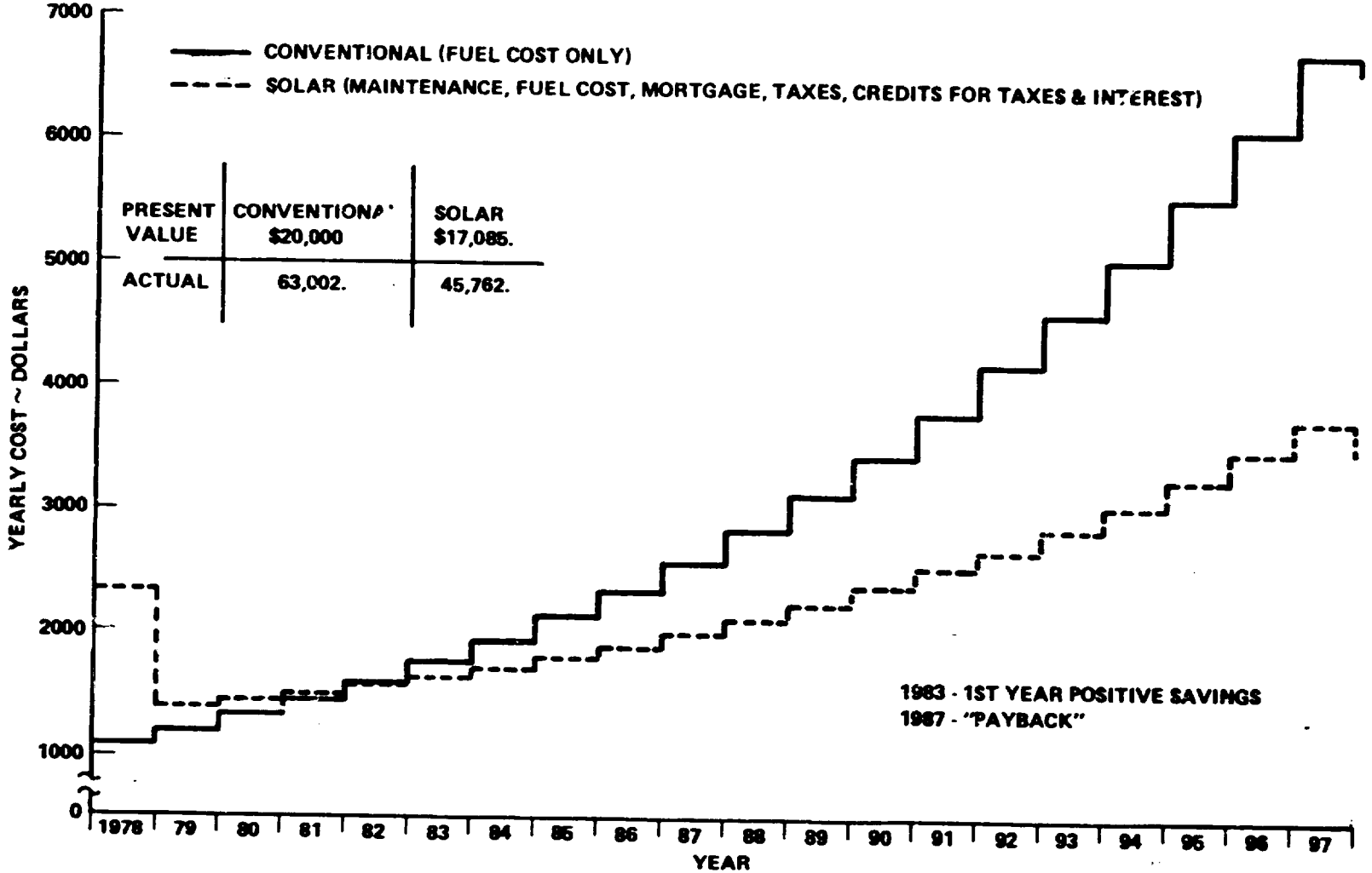


Figure 12. Sample problem cash flow.

TABLE 13. SAMPLE PROBLEM RESULTS

Solar System Life Cycle Costs	Present Values
Mortgage	$ 8,393.69
Property Tax	2,773.38
Maintenance	1,386.69
Downpayment	1,000.00
Conventional Fuel	6,000.00
Tax Credits	
Property Tax	$ -832.01
Interest	$-1,637.07
Total Cost	$17,084.68
Conventional System Life Cycle Cost	
Conventional Fuel	$20,000.00
Total Cost	$20,000.00
Life Cycle Savings ($20,000-$17,084.68)	$ 2,915.32
Conventional Fuel Savings	$14,000.00

A. Sensitivity Studies

In the discussion of Section II data input, six cards are described which contain the data input for: discount rate, inflation rate, down payment factor, property tax rate, income tax rate and collector area. To study the sensitivity of life cycle cost to any of these parameters, it is only necessary to list on each of these cards the starting value of the parameter, the ending value, then the increment by which the beginning value is increased in steps up to the ending value. A dash (-) separates the starting and ending values and an "X" separates the ending value and the increment. For example, to study in the sample problem the sensitivity of life cycle cost to collector area variations from 30 m^2 to 60 m^2 in increments of 5 m^2 the card should read "30.-60.X5." rather than "48.28" as in the sample problem. The summary output table will now contain the results of all seven cases. Table 14 is an example.

TABLE 14. SUMMARY OUTPUT TABLE

SAMPLE PROBLEM BIRMINGHAM, ALABAMA OCT. 2, 1979 16.0.100

SUMMARY

BIRMINGHAM, ALABAMA

DISCOUNT RATE = 10.00%
INFLATION RATE = 6.00%
DOWNPAYMENT = 10.00%
PROPERTY TAX RATE = 2.00%
INCOME TAX RATE = 30.00%
COST PER SQ. METER $165.73
FOR THIS CASE, VARY 1S THE COLLECTOR AREA. GJ/YR

ECT NO	VARY	PERCENT LOAD	TOTAL LOAD	ACTUAL COST SHVAC	ACTUAL COST CHVAC	PRESENT VAL COST SHVAC	PRESENT VAL COST CHVAC	LIFE CYCLE COST SAVINGS PRESENT	LIFE CYCLE COST SAVINGS ACTUAL	LIFE CYCLE COST SAVINGS ANNUAL	LIFE CYCLE FUEL COST PRESENT	LIFE CYCLE FUEL COST ACTUAL	LIFE CYCLE FUEL COST ANNUAL
1	30.0	63.07%	66.04	48292.11	63002.99	17418.10	20000.00	2006.02	18710.38	339.08	10613.95	33435.27	1296.71
2	35.0	58.52%	61.80	47085.02	63002.99	16991.90	20000.00	3058.10	15917.47	359.70	11703.60	36867.80	1374.73
3	40.0	63.31%	66.85	46294.13	63002.99	16902.73	20000.00	3097.27	16708.36	363.60	12661.13	39884.13	1487.17
4	45.0	67.52%	71.30	45867.05	63002.99	16979.06	20000.00	3020.94	17135.44	354.84	13603.17	42536.66	1586.09
5	50.0	71.22%	75.21	45758.20	63002.99	17156.91	20000.00	2843.09	17244.29	334.01	14294.18	44870.96	1673.12
6	55.0	74.48%	78.65	45927.90	63002.99	17422.18	20000.00	2577.82	17074.59	302.79	14896.77	46926.69	1749.77
7	60.0	77.36%	81.69	46341.61	63002.99	17765.42	20000.00	2234.58	16660.88	262.47	15471.90	48738.43	1817.32

B. Plot Capability

An x-y coordinate system plot capability is provided to display the results of a sensitivity study. In the Section II data input shown in Figure 6, there is a zero in Column 75. This indicates no plot is desired. When a sensitivity study is done for one of the six variables just described, then plots may be desired. Table 15 defines what plots result from inserting either a 1, 2 or 3 in place of the zero in Column 75. Figures 13 and 14 are examples of plots for the sample problem with collector area as the sensitivity parameter. Only one of the six sensitivity variables can be plotted in a single execution of the program.

C. Listing of Input Data

If a listing of the input data is not required, insert a "1" in Column 55 of the first card in Section II. See Section III.A.2. for the discussion of inputing the first card in Section II.

D. Life Cycle Savings

Life cycle savings will not be computed if a zero is inserted in Column 65 of the first card of Section II input.

E. When to Inflate

Costs in the first year of the analysis will not be inflated if a "1" is input in Column 70 of the first card of Section II input. Normally, i.e., with a zero in Column 70, first year costs are inflated. Whether or not first year costs are inflated is a matter of convention.

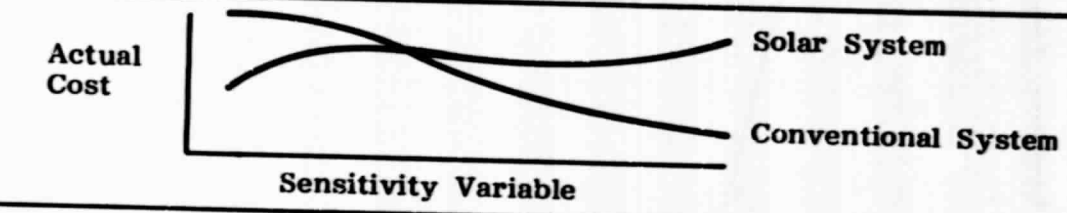
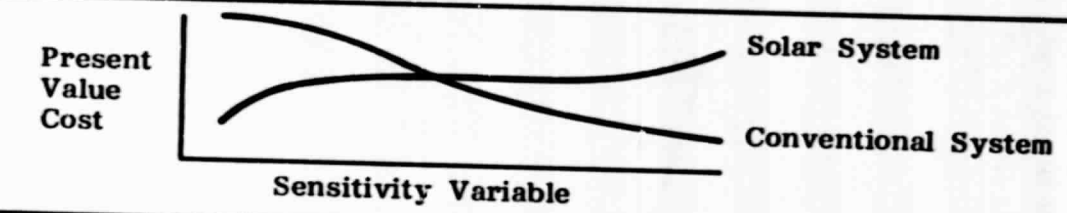
F. Tax Credits for Energy Conservation

Federal and State tax credits for energy conservation effectively reduce initial system cost. Therefore, in the user problem, the amount of credit can be deducted from acquisition cost before input to the program.

G. Collector Performance Parameters

The user may desire to use different values of $F'_R\tau\alpha$ and $F'_R U_L$ in the GFL method or the percent solar fraction may be known. To use different collector performance parameters, it is necessary to insert one

TABLE 15. PLOT RESULTS FOR 0, 1, 2, AND 3 IN COLUMN 75

Value in Column 75	Plot Obtained
0	No Plots
1	
2	
3	Plots of both actual cost and present value cost, i.e., equivalent to "1" and "2"

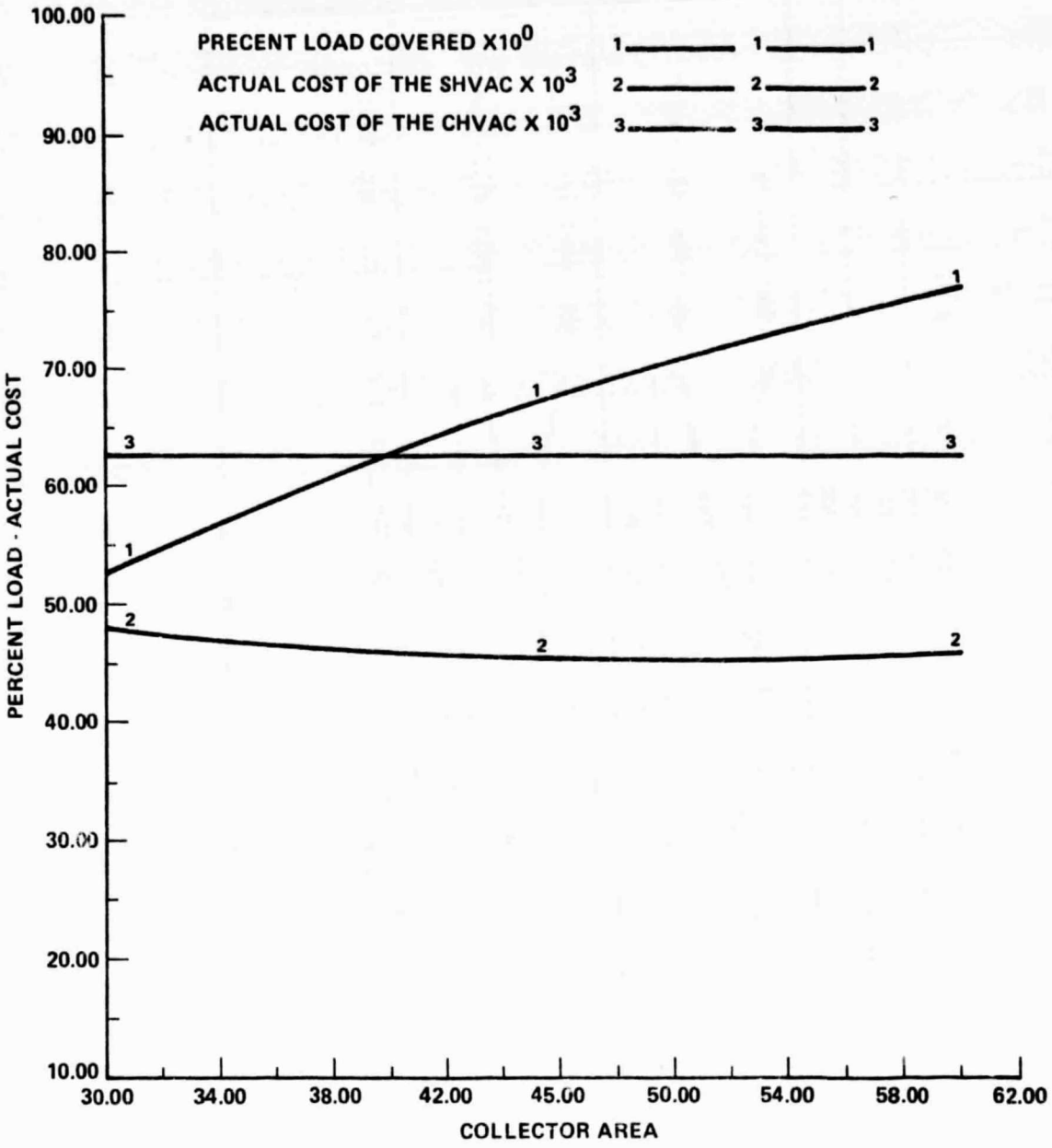


Figure 13. Performance and cost as related to collector area.

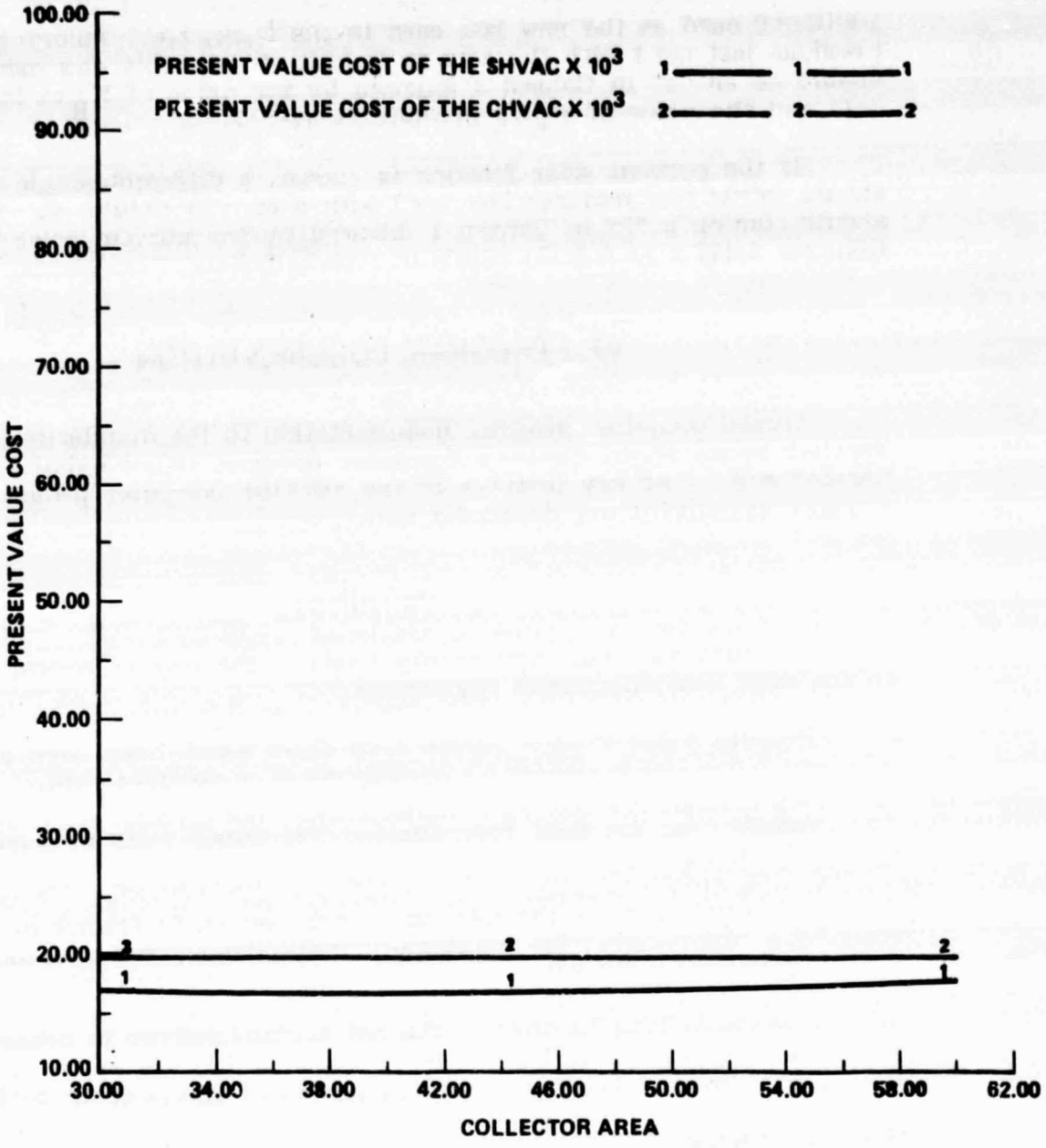


Figure 14. Present value cost as related to collector area.

additional card as the new last card in the deck, i.e., following the previous last card with the zero in Column 4. On this new card there should be an "F" in Column 1 followed by the value of $F'_R\tau\alpha$ in Columns 2-11 and the value of $F'_R U_L$ in Columns 12-21.

If the percent solar fraction is known, a different single new card should follow the previous last card with a zero in Column 4. This card should contain a "P" in Column 1 followed by the percent solar in Columns 2-11.

VI. PROGRAM DOCUMENTATION

Formal computer program documentation to the standards of the Marshall Space Flight Center's Computer Services Office exists in Reference 8. The key features of the SHCOST computer program from a user's standpoint are described here.

A. Definitions

Discount Rate (or Opportunity Cost) — the rate of return foregone on the next best alternative investment.

Present Value Cost — future cash flows which have been converted to present day dollar values by application of a discount rate.

Actual Cost (or Real Year Cost) — the dollar value of a cost at the time it is incurred.

Life Cycle Cost — the total of all relevant costs (either in present value of actual dollars) for acquisition, maintenance, and operation of a system over its useful life.

Life Cycle Cost Savings — the net savings (either in present value or actual dollars) obtained by operating a solar heating system rather than a conventional system during the period of analysis; or mathematically, the life cycle cost of the conventional system minus the life cycle cost of the solar system.

B. Equations

The following terms may apply to either the solar heating system or the conventional system, depending on the user problem. The basic mathematical expression for present value (PV) is defined.

Acquisition

$$PV = I$$

where

I = initial acquisition cost incurred for purchase, delivery, integration and installation of the system. With debt financing I becomes simply the downpayment.

Building Modifications

$$PV = BM$$

where

BM = initial, one-time cost for building modifications.

Net Replacement and Repair

$$PV = \sum_{j=1}^{N} \frac{NRR_j}{(1+d)^j}$$

where

NRR_j = net replacement and repair cost in year j.

j = 1 for first year of analysis and j = N for last year

N = period of analysis in years

d = annual discount rate

Maintenance

$$PV_M = M \sum_{j=1}^{N} \left(\frac{1+i}{1+d} \right)^j$$

where

M = maintenance cost in the first year of system life

i = general inflation rate

Conventional Fuel Cost

$$PV_{CFC} = F \sum_{j=1}^{N} \left(\frac{1 + e}{1 + d} \right)^{j}$$

where

F = conventional fuel cost in the first year

e = conventional fuel price escalation rate

Property Taxes

$$PV_{PT} = tI \sum_{j=1}^{N} \left(\frac{1 + i}{1 + d} \right)^{j}$$

where

I = initial total system acquisition cost

t = property tax rate which when multiplied by I gives first year taxes

Added Income

$$PV = -(1-\bar{t})Y \frac{(1+d)^{N} - 1}{d(1+d)^{N}}$$

where

$\bar{t}$ = income tax rate

Y = annual gross revenue

Insurance

$$PV = P \frac{(1 + d)^{N} - 1}{d(1+d)^{N}}$$

where

P = first year insurance premium

Salvage

$$PV = - \frac{S}{(1+d)^N}$$

where

S = salvage value of system at end of period of analysis

Loan Payments

$$PV = PMT \frac{(1 + d)^N - 1}{d(1+d)^N}$$

where

PMT = yearly mortgage payment

Property Tax Credits

$$PV = - PV_{PT} \bar{t}$$

where

PV_{PT} = present value of property taxes

Maintenance Expense Credits

$$PV = - PV_M \bar{t}$$

where

PV_M = present value of maintenance expenses

Depreciation Credit

$$PV = \frac{\bar{t} (I-S)}{N} \frac{(1+d)^N - 1}{d(1+d)^N}$$

Loan Interest Credit

$$PV = \bar{t} \sum_{j=1}^{N} \frac{MB_j \ell}{(1+d)^j}$$

where

MB_j (mortgage balance) = $MB_{j-1} - (PMT - MB_{j-1}\ell)$

ℓ = loan interest rate

Conventional Fuel Cost Credit

$$PV = - PV_{CFC}\, \bar{t}$$

where

PV_{CFC} = present value of conventional fuel cost

C. Flow Chart

A flow chart of the program is presented in Figure 15 and a brief description of the function of each subroutine follows Figure 15.

D. Listing

A complete program listing is presented in Appendix B.

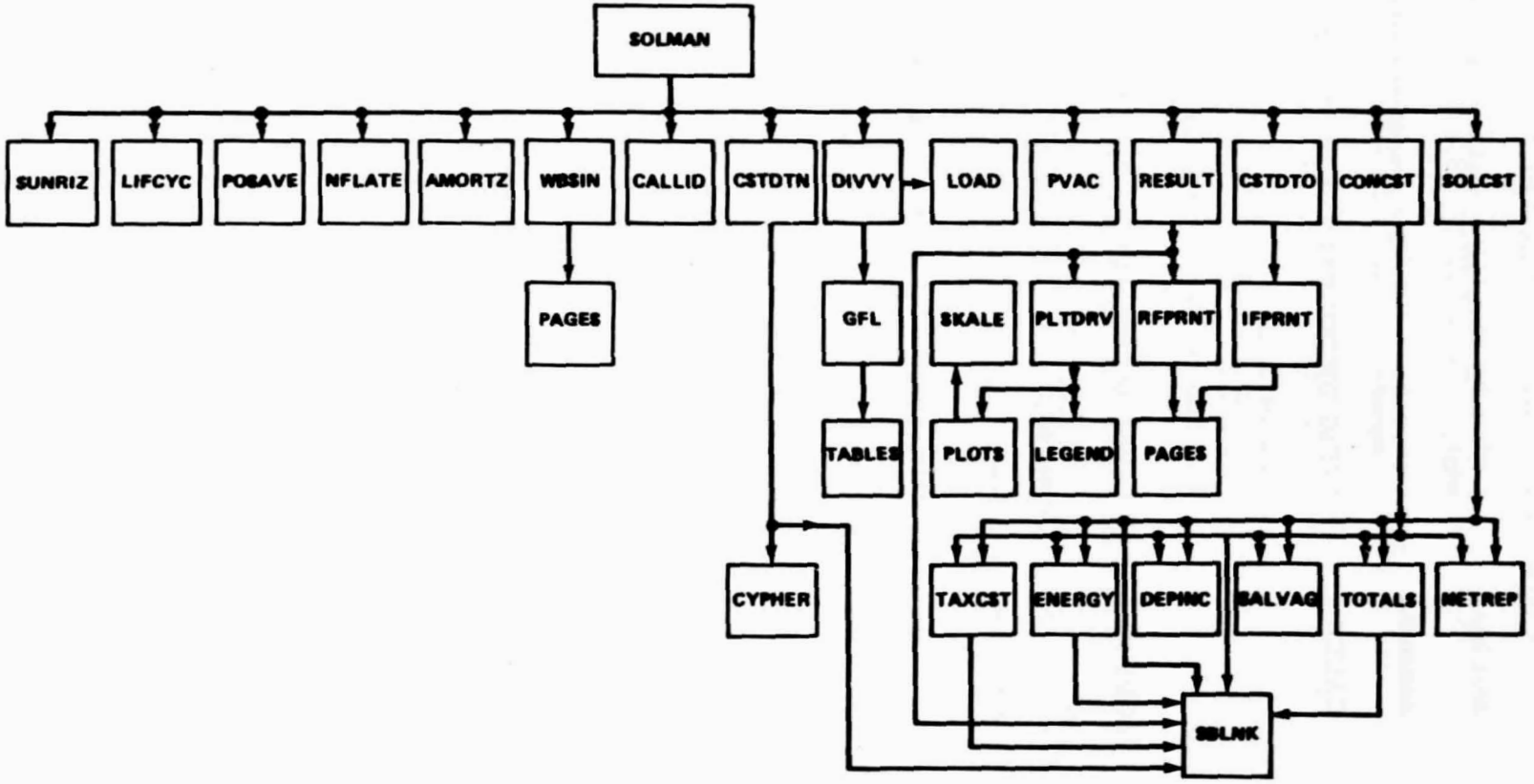


Figure 15. SHCOST flow chart.

ACCNO	Called by CALLID. Gets runs account number for plot.
AMORTZ	Called by SOLMAN. Computes the annual load payment.
CALLID	Called by SOLMAN. Calls system routine projid to get the run's project ID and accno to get the run's account number.
CONCST	Called by SOLMAN. Calls SBLNK, DEPINC, SALVAG, TOTALS NETREP, ENERGY and TAXCST. Computes cost for conventional system.
CSTDTN	Called by SOLMAN, calls CYPHER. Reads cost inputs.
CSTDTO	Called by SOLMAN, calls IFPRNT. Displays cost inputs.
CYPHER	Called by CSTDTN. Reads and decodes cost inputs.
DEPINC	Called by CONCST and SOLCST. Computes deduction for commercial owner. Depreciation, rental, operation expenses and maintenance figure in computations.
DIVVY	Called by SOLMAN. Calls GFL and LOAD. Initiates each case computation.
ENERGY	Called by CONCST and SOLCST. Computes energy cost. Cost for each type of fuel or energy source per year are escalated here.
LIFCYC	Called by SOLMAN. Computes life cycle cost.
IFPRNT	Called by CSTDTO. Prints cost inputs.
GFL	Called by DIVVY. Calls TABLE. Sizes solar system.
NETREP	Called by CONCST, SOLCST. Computes net replacement cost and maintenance cost.
LEGEND	Called by PLTDRV. Displays plot labels on CALCOMP.
NFLATE	Called by SOLMAN. Inflates cost input.
PLOTS	Called by PLTDRV, calls SKALE. Plots data, draws grid.
PLTDRV	Called by RESULT, calls PLOTS, LEGEND. Establishes labels for plots.
POSAVE	Called by SOLMAN. Computes which year positive savings will occur and by how much.
PROJID	Called by CALLID to get runs project ID for plot.
PVAC	Called by SOLMAN. Totals yearly results

Figure 15. (Continued)

RESULT	Called by SOLMAN, calls RFPRNT, SBLNK, PLTDRV. Prints calculated results, generates plot data.
RFPRNT	Called by RESULT. Prints result data.
SALVAG	Called by CONCST, SOLCST. Computes SALVAG values.
SBLNK	Called by CONCST, SOLCST, RESULT, PVAC. Clears memory.
SKALE	Called by PLOTS. Scales plot data.
SOLCST	Called by SOLMAN, calls SBLNK, DEPINC, SALVAG, TOTALS, NETREP, ENERGY, TAXCST. Computes cost for solar system.
SOLMAN	Main routine, calls CALLID, WBSIN, CSTDTN, DIVVY, AMORTZ, NFLATE, CSTDTO, LIFCYC, PVAC, RESULT, POSAVE, SUNRIZ, CONCST, SOLCST.
SUNRIZ	Called by SOLMAN. Displays a summary of every case.
TABLE	Called by GFL. Contains data for each sight, liquid and air solar system.
TAXCST	Called by CONCST, SOLCST. Computes total cost result for each year of each cost input, and adds those to total cost.
WBSIN	Called by SOLMAN. Read data for WBS dictionary structure.
LOAD	Called by DIVVY. Loads cost data for storage.

Figure 15. (Concluded)

VII. REFERENCES

1. "An Analysis of the Current Economic Feasibility of Solar Water and Space Heating," Office of the Assistant Secretary for Conservation and Solar Applications, U. S. Department of Energy, Washington, D. C., November 1977.

2. "A Design Procedure for Solar Heating Systems," S. A. Klein, W. A. Beckman, J. A. Duffie, Solar Energy, Col. 18, pp. 113-127, 1976.

3. "Solar Heating and Cooling in Buildings: Methods of Economic Evaluation," Rosalie T. Ruegg, NBSIR 75-712, National Bureau of Standards, July 1975.

4. ASHRAE Handbook - 1977 Fundamentals, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 345 East 47th St., New York, N. Y. 10017.

5. "The GFL Method for Sizing Solar Energy Space and Water Heating Systems," G. F. Lameiro, P. Bendt, Solar Energy Research Institute, SERI-30, May 1978.

6. "Handbook for Preparation of Work Breakdown Structures," NHB 5610.1, National Aeronautics and Space Administration, February 1975.

7. Principles of Engineering Economy, 5th ed. (New York: The Ronald Press Co., 1970).

8. "SHCOST - Solar Heating, Ventilation and Cooling Cost Analysis Program - Version II," prepared for NASA/MSFC Huntsville Computer Complex by Computer Sciences Corporation, Aerospace Systems Center, April 20, 1979.

APPENDIX A

SAMPLE PROBLEM

```
@SRG,S

SRG-0001N SL75R1 10/01/79 20:36:10

SH*COST,1R532
CALL TO SECURE NOT NECCESSARY ALL FILES ARE CATALOGUED


@ADD,P  ASSIGN.

@ASG,A              SH*COST.

@XQT SH*COST.SHCOST
        1      SAMPLE PROBLEM
        2       11.0  SOLAR SYSTEM LC COST      2  6  7 15 14  8  9
        3       21.1  ACQUISITION              18 19
        4      181.1.1COLLECTOR
        5      191.1.2OTHER COST ELEMENTS
        6       61.2  MAINTENANCE
        7       71.3  CONV. FUEL COST
        8      151.4  LOAN PAYMENTS
        9      141.5  LOAN INTEREST CREDITS
       10       81.6  PROPERTY TAXES
       11       91.7  PROPERTY TAX CREDITS
       12     1012.0  CONV. SYSTEM LC COST    107
       13     1072.1  CONV. FUEL COST
       14
       15         20 1980     165.70   18   16    1      105.6    0    0    0    1    0    0
       16     0.1
       17     0.06
       18     0.1
       19     0.02
       20     0.3
       21     48.28
       22     6
       23     .01
       24     7
       25     8
       26     9
       27     15
       28     14
       29     20.
       30     .09
       31     19
       32     2000.00
       33     107
       34     1000.00
       35     .1
       36        0
NORMAL EXIT     CPU:        347        SUPS:        2987      (MILLISECONDS)

@XQT SH*COST.COST
```

WBS DICTIONARY

1	1.0	SOLAR SYSTEM LC COST
2	1.1	ACQUISITION
18	1.1.1	COLLECTOR
19	1.1.2	OTHER COST ELEMENTS
6	1.2	MAINTENANCE
7	1.3	CONV. FUEL COST
15	1.4	LOAN PAYMENTS
14	1.5	LOAN INTEREST CREDITS
8	1.6	PROPERTY TAXES
9	1.7	PROPERTY TAX CREDITS
101	2.0	CONV. SYSTEM LC COST
107	2.1	CONV. FUEL COST

THIS HEATING VENTILATION AND COOLING ANALYSIS
WILL COVER 20 YEARS BEGINNING WITH 1980 AND
ENDING WITH 1999
THE DISCOUNT RATE TO BE APPLIED TO THE ANALYSIS IS --- 10.00%
THE INFLATION FACTOR TO BE APPLIED TO THE ANALYSIS IS- 6.00%

LISTING OF COST DATA INPUT

1.0 SOLAR SYSTEM LC COST
- 1.1 ACQUISITION
- 1.2 MAINTENANCE
- 1.3 CONV. FUEL COST
- 1.4 LOAN PAYMENTS
- 1.5 LOAN INTEREST CREDITS
- 1.6 PROPERTY TAXES
- 1.7 PROPERTY TAX CREDITS

1.1 ACQUISITION

1.1.1 COLLECTOR

ANNUAL LOAD =	105.60 GJ/YR
PERCENT LOAD COVERED =	70.00%
INITIAL COST =	8000.00

1.1.2 OTHER COST ELEMENTS

INITIAL COST =	2000.00

DOWNPAYMENT WILL BE 10.00% OF TOTAL COST

1.2 MAINTENANCE

COST PER YEAR =	100.00

1.3 CONV. FUEL COST

ANNUAL FUEL COST =	300.01
ESCALATION FACTOR =	10.00%

1.4 LOAN PAYMENTS

MARKET INTEREST RATE = .00%
ANNUAL PAYMENT = 985.92

1.5 LOAN INTEREST CREDITS

MARKET INTEREST RATE = 9.00%
INCOME TAX RATE = 30.00%

YEAR	OUTSTANDING MORTGAGE VALUE
1980	9000.00
1981	8824.08
1982	8632.33
1983	8423.32
1984	8195.50
1985	7947.18
1986	7676.51
1987	7381.47
1988	7059.89
1989	6709.36
1990	6327.28
1991	5910.82
1992	5456.88
1993	4962.08
1994	4422.75
1995	3834.88
1996	3194.10
1997	2495.65
1998	1734.34
1999	904.51

1.6 PROPERTY TAXES

PROPERTY TAX RATE = 2.00%
ASSESSED VALUE = 10000.00

1.7 PROPERTY TAX CREDITS

PROPERTY TAX RATE = 2.00%
INCOME TAX RATE = 30.00%
ASSESSED VALUE = 10000.00

****

LISTING OF COST DATA INPUT

2.0 CONV. SYSTEM LC COST
2.1 CONV. FUEL COST

2.1 CONV. FUEL COST

ANNUAL FUEL COST = 1000.00
ESCALATION FACTOR = 10.00%

1.0 RESULTS FOR SOLAR SYSTEM LC COST

1.1 ACQUISITION

COSTS OVER THE 20 YEARS OF THE ANALYSIS

THE DOWN PAYMENT = 1000.01
10.00% OF THE TOTAL COST

1.2 MAINTENANCE

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	1386.69
ANNUAL COST	=	162.88
ACTUAL COST	=	3899.27

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	96.36	106.00
1981	92.86	112.36
1982	89.46	119.10
1983	86.23	126.25
1984	83.09	133.82
1985	80.07	141.85
1986	77.16	150.36
1987	74.35	159.38
1988	71.65	168.95
1989	69.04	179.08
1990	66.53	189.83
1991	64.11	201.22
1992	61.78	213.29
1993	59.54	226.09
1994	57.37	239.66
1995	55.29	254.04
1996	53.28	269.28
1997	51.34	285.43
1998	49.47	302.56
1999	47.67	320.71

1.3 CONV. FUEL COST

COSTS OVER THE 20 YEARS OF THE ANALYSIS
PRESENT VALUE = 6000.15
ANNUAL COST = 704.78
ACTUAL COST = 18901.21

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	300.01	330.01
1981	300.01	363.01
1982	300.01	399.31
1983	300.01	439.24
1984	300.01	483.16
1985	300.01	531.48
1986	300.01	584.63
1987	300.01	643.09
1988	300.01	707.40
1989	300.01	778.14
1990	300.01	855.96
1991	300.01	941.55
1992	300.01	1035.71
1993	300.01	1139.28
1994	300.01	1253.21
1995	300.01	1378.53
1996	300.01	1516.38
1997	300.01	1668.02
1998	300.01	1834.82
1999	300.01	2018.30

1.4 LOAN PAYMENTS

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	8393.68
ANNUAL COST	=	985.92
ACTUAL COST	=	19718.36

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	896.29	985.92
1981	814.81	985.92
1982	740.73	985.92
1983	673.40	985.92
1984	612.18	985.92
1985	556.52	985.92
1986	505.93	985.92
1987	459.94	985.92
1988	418.13	985.92
1989	380.11	985.92
1990	345.56	985.92
1991	314.14	985.92
1992	285.59	985.92
1993	259.62	985.92
1994	236.02	985.92
1995	214.56	985.92
1996	195.06	985.92
1997	177.33	985.92
1998	161.21	985.92
1999	146.55	985.92

1.5 LOAN INTEREST CREDITS

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	1637.07
ANNUAL COST	=	192.29
ACTUAL COST	=	3215.51

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	220.91	243.00
1981	196.90	238.25
1982	175.11	233.07
1983	155.34	227.43
1984	137.40	221.28
1985	121.12	214.57
1986	106.36	207.27
1987	92.97	199.30
1988	80.84	190.62
1989	69.84	181.15
1990	59.88	170.84
1991	50.85	159.59
1992	42.68	147.34
1993	35.28	133.98
1994	28.59	119.41
1995	22.53	103.54
1996	17.06	86.24
1997	12.12	67.36
1998	7.66	46.83
1999	3.63	24.42

1.6 PROPERTY TAXES

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	2773.38
ANNUAL COST	=	325.76
ACTUAL COST	=	7798.54

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	192.73	212.00
1981	185.72	224.72
1982	178.97	238.20
1983	172.46	252.50
1984	166.19	267.65
1985	160.14	283.70
1986	154.32	300.73
1987	148.71	318.77
1988	143.30	337.90
1989	138.09	358.17
1990	133.07	379.66
1991	128.23	402.44
1992	123.57	426.59
1993	119.07	452.18
1994	114.74	479.31
1995	110.57	508.07
1996	106.55	538.55
1997	102.68	570.87
1998	98.94	605.12
1999	95.34	641.43

1.7 PROPERTY TAX CREDITS

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	832.01
ANNUAL COST	=	97.73
ACTUAL COST	=	2339.56

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	57.82	63.60
1981	55.72	67.42
1982	53.69	71.46
1983	51.74	75.75
1984	49.86	80.29
1985	48.04	85.11
1986	46.30	90.22
1987	44.61	95.63
1988	42.99	101.37
1989	41.43	107.45
1990	39.92	113.90
1991	38.47	120.73
1992	37.07	127.98
1993	35.72	135.65
1994	34.42	143.79
1995	33.17	152.42
1996	31.97	161.57
1997	30.80	171.26
1998	29.68	181.54
1999	28.60	192.43

1.0 SOLAR SYSTEM LC COST

THE PRESENT VALUE COST OF THE SYSTEM* IS-- 17084.82

THE ANNUALIZED COST OF THE SYSTEM IS----- 2035.78

THE ACTUAL COST OF THE SYSTEM IS -------- 45762.32

COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	2206.67	2327.34
1981	1140.78	1380.34
1982	1080.39	1438.00
1983	1025.01	1500.72
1984	974.21	1568.98
1985	927.58	1643.27
1986	884.76	1724.15
1987	845.42	1812.23
1988	809.25	1908.18
1989	775.99	2012.71
1990	745.37	2126.63
1991	717.18	2250.80
1992	691.19	2386.19
1993	667.24	2533.84
1994	645.13	2694.88
1995	624.72	2870.59
1996	605.86	3062.32
1997	588.42	3271.59
1998	572.29	3500.05
1999	557.34	3749.51

2.0 RESULTS FOR CONV. SYSTEM LC COST

2.1 CONV. FUEL COST

COSTS OVER THE 20 YEARS OF THE ANALYSIS

PRESENT VALUE	=	20000.00
ANNUAL COST	=	2349.19
ACTUAL COST	=	63002.49

PRESENT VALUE AND ACTUAL COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	1000.00	1100.00
1981	1000.00	1210.00
1982	1000.00	1331.00
1983	1000.00	1464.10
1984	1000.00	1610.51
1985	1000.00	1771.56
1986	1000.00	1948.72
1987	1000.00	2143.59
1988	1000.00	2357.95
1989	1000.00	2593.74
1990	1000.00	2853.12
1991	1000.00	3138.43
1992	1000.00	3452.27
1993	1000.00	3797.50
1994	1000.00	4177.25
1995	1000.00	4594.97
1996	1000.00	5054.47
1997	1000.00	5559.92
1998	1000.00	6115.91
1999	1000.00	6727.50

2.0 CONV. SYSTEM LC COST

THE PRESENT VALU COST OF THE SYSTEM' IS-- 20330.00
THE ANNUALIZED COST OF THE SYSTEM IS----- 2349.19
THE ACTUAL COST OF THE SYSTEM IS -------- 63002.49

COSTS BY YEAR

YEAR	PVCOST	ACTUAL
1980	1000.00	1100.00
1981	1000.00	1210.00
1982	1000.00	1331.00
1983	1000.00	1464.10
1984	1000.00	1610.51
1985	1000.00	1771.56
1986	1000.00	1948.72
1987	1000.00	2143.59
1988	1000.00	2357.95
1989	1000.00	2593.74
1990	1000.00	2853.12
1991	1000.00	3138.43
1992	1000.00	3452.27
1993	1000.00	3797.50
1994	1000.00	4177.25
1995	1000.00	4594.97
1996	1000.00	5054.47
1997	1000.00	5559.92
1998	1000.00	6115.91
1999	1000.00	6727.50

LIFE CYCLE COST SAVINGS

PRESENT	=	2915.18
ACTUAL	=	17293.17
ANNUALIZED	=	342.42
THE PRESENT VALUE FUEL SAVINGS	=	13999.85
THE ACTUAL FUEL SAVINGS	=	44101.28
THE ANNUALIZED FUEL SAVINGS	=	1644.42

IN YEAR 1985 POSITIVE SAVINGS WILL OCCUR BY
AN AMOUNT OF 25.79 IN TERMS OF PRESENT VALUE DOLLARS
IN YEAR 1985 POSITIVE SAVINGS WILL OCCUR BY
AN AMOUNT OF 41.53 IN TERMS OF ACTUAL DOLLARS

CUMULATIVE SAVINGS BY YEAR

YEAR	PRESENT VALUE SAVINGS	ACTUAL SAVINGS
1980	-1206.67	-1227.34
1981	-1347.45	-1397.68
1982	-1427.84	-1509.68
1983	-1452.85	-1541.30
1984	-1427.06	-1499.77
1985	-1359.65	-1371.48
1986	-1239.41	-1196.91
1987	-1084.83	-815.56
1988	-894.08	-365.79
1989	-670.07	215.24
1990	-415.44	941.73
1991	-132.62	1829.36
1992	176.19	2895.44
1993	508.95	4159.10
1994	863.82	5641.46
1995	1239.09	7365.85
1996	1633.23	9358.00
1997	2044.80	11646.32
1998	2472.52	14262.18
1999	2915.18	17240.17

SUMMARY

BIRMINGHAM, ALABAMA

DISCOUNT RATE	= 10.00%
INFLATION RATE	= 6.00%
DOWNPAYMENT	= 10.00%
PROPERTY TAX RATE	= 2.00%
INCOME TAX RATE	= 30.00%
COST PER SQ. METER	$165.70

FOR THIS CASE, VARY IS THE COLECTOR AREA. GJ/YR

SECT NO	VARY	PERCNT LOAD	TOTAL LOAD	ACTUAL COST SHVAC	ACTUAL COST CHVAC	PRESENT VAL COST SHVAC	PRESENT VAL COST CHVAC	LIFE CYCLE COST SAVINGS PRESENT	LIFE CYCLE COST SAVINGS ACTUAL	LIFE CYCLE COST SAVINGS ANNUAL	LIFE CYCLE FUEL COST PRESENT	LIFE CYCLE FUEL COST ACTUAL	LIFE CYCLE FUEL COST ANNUAL
1	48.3	70.00%	73.92	45762.32	63002.39	17084.82	20000.00	2915.18	17240.17	342.42	13999.85	44101.28	1644.4

NORMAL EXIT CPU: 2436 SUPS: 4 8472 (MILLISECONDS)

*FIN

APPENDIX B

SOURCE LIST

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890**SEQ NO. 00


$HDS*P SHCOST SOURCE LIST

$PRT*S A.AMORTZ
FURPUR 27R3AH4     E33 SL73R1 09/28/79 12:58:36
```

```
SH*COST(1),AMORTZ
     1              SUBROUTINE AMORTZ
     2               INCLUDE PARS
     3              IF(IRC1+IRC2.EQ.0) GO TO 200
     4              READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
     5              READ(8'IRC2,ERR=600) ISA,ACODE,ATITLE,ACOST
     6        C*** COMPUTE THE AMORTIZED PAYMENTS AND PLACES OUTSTANDING
     7        C*** MORTGAGE ON WHICH INTEREST IS PAID IN YEAR (I) IN COST
     8                   A=COST(1)
     9                   N=COST(2)+0.5
    10                   COST(NYR+1)=COST(3)
    11              ACOST(NYR+1)=ACOST(3)
    12              ACOST(NYR+2)=ACOST(4)
    13                   COST(NYR+2)=COST(4)
    14        C*** COMPUTE ANNUAL PAYMENTS
    15                   PAY=A*(COST(NYR+1)/(1.0-(1.0+COST(NYR+1))**(-N)))
    16              ACOST(1)=PAY
    17                   DO 100 K=2,NYR
    18              ACOST(K)=PAY
    19                   KK=K-1
    20                   COST(K)=COST(KK)-(PAY-COST(NYR+1)*COST(KK))
    21                   IF(COST(K) .LT.  1.0)COST(K)=0.0
    22        100        CONTINUE
    23              WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
    24              WRITE(8'IRC2,ERR=600) ISA,ACODE,ATITLE,ACOST
    25        200     RETURN
    26        600   WRITE(6,601)
    27        601   FORMAT(5X,'ERROR IN AMORTZ')
    28              RETURN
    29                   END
DATA IGNORED - IN CONTROL MODE

APRT,S A.CALLID
FURPUR 27R34H9     E33 SL73R1 09/28/79 12:58:40
```

```
SH*COST(1).CALLID
     1              SUBROUTINE CALLID
     2              CHARACTER*12 N
     3              CHARACTER*6 N1,N2,E,F
     4              CALL PLOT(0.0,0.0,-3)
     5              CALL PLOT(9.0,10.0,2)
     6              CALL PLOT(8.5,10.0,2)
     7              CALL PLOT(8.5,0.0,2)
     8              CALL PLOT(0.0,0.0,2)
     9              CALL PROJID(N)
    10              DECODE(12,100,N) N1,N2
    11       100    FORMAT(2A6)
    12              CALL SYMBOL(2.3,8.0,.286,N1,0.0,6)
    13              CALL SYMBOL(5.6,8.0,.286,N2,0.0,6)
    14              CALL ACCNO(N)
    15              DECODE(12,100,N) N1,N2
    16              CALL SYMBOL(2.3,6.0,.286,N1,0.0,6)
    17              CALL SYMBOL(5.6,6.0,.286,N2,0.0,6)
    18              CALL SCLOCK(N1,N2,E,F)
    19              CALL SYMBOL(2.3,4.0,.286,N1,0.0,6)
    20              CALL SYMBOL(5.6,4.0,.286,N2,0.0,6)
    21              CALL PLOT(13.5,0.0,-3)
    22              RETURN
    23              END
DATA IGNORED - IN CONTROL MODE

@PRT,S A.CONCST
FURPUR 27R3AH4    E33 SL73R1 09/28/79 12:58:44
```

```
SH•COST(1).CONCST
      1             SUBROUTINE CONCST
      2             INCLUDE PARS
      3       C   COST DRIVER FOR COSTING CONVENTIONAL SYSTEM
      4             IRC1=102
      5       C••• TOTAL COSTS INTO UPPER LEVEL
      6             CALL TOTALS
      7             IRC1=103
      8             CALL TOTALS
      9             IRC1=104
     10             CALL TOTALS
     11       C  NETREPLACEMENTS
     12             IRC1=105
     13             IRC2=106
     14             CALL NETREP
     15             IRC1=107
     16       C••• COMPUTE ENERGY COSTS BY YEAR FOR PV AND FOR EACH SUBLEVEL
     17             CALL ENERGY
     18             IRC1=109
     19             IRC2=108
     20             IRC3=111
     21             IRC4=113
     22             IRC5=114
     23       C••• COMPUTE COSTS OF TAXES AND DEDUCTIONS FROM TAXES
     24             CALL TAXCST
     25             IRC1=113
     26             IRC2=0
     27       C••• COMPUTE DEPRECIATION DEDUCTIONS AS ADDED INCOME
     28             IRC3=115
     29             IRC4=116
     30             IF(ICOM.EQ.1) CALL DEPINC
     31             IRC1=112
     32       C••• COMPUTE SALVAGE VALUE AFTER N YEARS OF ANALYSIS•••
     33             CALL SALVAG
     34       100   IRC1=101
     35       C••• ADD ALL SUBLEVELS FOR THE TOTAL PV AND AC
     36             CALL PVAC
     37             RETURN
     38             END
DATA IGNORED - IN CONTROL MODE

?PRT•S A•CSTDTN
FURPUR 27R3AM4     E33 SL73R1 09/28/79 12:58:51
```

SH*COST(1).CSTDTN

```
      1              SUBROUTINE CSTDTN
      2              INTEGER CNT
      3              COMMON/NFLT/FLATE
      4              COMMON/GFL2/ XSTART,XSTOP,XSTEP
      5              INTEGER TYPE
      6              INCLUDE PARS
      7              COMMON/INPUTS/COSTIN(200,3)
      8        C    READ DATA IN FOR EACH WBS BLOCK WHICH  REQUIRES COST DATA
      9        C    READ NYR START YEAR AND DISCOUNT RATE
     10        C     DISCOUNT RATE DISC SHOULD BE INPUT AS DECIMAL FRACTION NOT AS %
     11        C     ICOM = 1 IF HVACS ARE COMMERCIAL ELSE ICOM = 0
     12        C     CHECK NYR AGAINST IYR IN PROC PARS BEFORE XQT
     13        C     DATOPT = 1 FOR COST DATA OUTPUT SUPPRESSION
     14        C***   FLATE=INFLATION RATE EXPRESSED AS FRACTION
     15        C***   IDSC=0 INDICATES DISCOUNT RATE IS NOMINAL(MARKET) RATE AND
     16        C***   DISC WILL HAVE THE INFLATION RATE SUBTRACTED FROM IT
     17        C***   IDSC NOT = 0 INDICATES DISC IS REAL RATE
     18        C***   INRGY=1 INDICATES ANALYSSI IS COMPARISON OF ENERGY COSTS AND IC1
     19        C***   AND IC2 CONTAIN  COSTS ALREADY COMPUTED TO BE USED FOR SYSTEM COSTS
     20        C***   IN LIFE CYCLE COSTS
     21        C***   ILFCY=1 INDICATES LIFE CYCLE COSTS ARE REQUESTED
     22        C***   IDISC=1 INDICATES  INFLATION RATE IS TO BE APPLIED
     23        C***   AFTER THE FIRST YEAR....IDISC/=/ 1 INDICATE FIRST YEAR INFLATE
     24        C****** READ NYR START YEAR DISCOUNT AND INFLATION FACTORS
     25        100    CONTINUE
     26               READ(5,110) NYR,IYRST,CSTPSR,CLTPRR,LCN,LOA,XLOAD
     27              +,ICOM,DATOPT,IDSC,ILFCY,IDISC
     28              *,IPLOTS
     29               CALL CYPHER(DSSTRT,DSSTOP,DSSTEP)
     30               CALL CYPHER(FTSTRT,FTSTOP,FTSTEP)
     31               CALL CYPHER(DPSTRT,DPSTOP,DPSTEP)
     32               CALL CYPHER(PTSTRT,PTSTOP,PTSTEP)
     33               CALL CYPHER(TSTART,TSTOP,TSTEP)
     34               CALL CYPHER(XSTART,XSTOP,XSTEP)
     35        C      DS  =  DISC   -  DISCOUNT RATE
     36        C      FT  =  FLATE  -  INFLATION RATE
     37        C      DP  =  DWNPAY  -  DOWNPAYMENT
     38          110 FORMAT(2I5,F10.0,3I5,F10.0,7I5)
     39               IPLTOP=1
     40               IF(DSSTEP.NE.0.0) IPLTOP=2
     41               IF(FTSTEP.NE.0.0) IPLTOP=3
     42               IF(DPSTEP.NE.0.0) IPLTOP=4
     43               IF(PTSTEP.NE.0.0) IPLTOP=5
     44               IF(TSTEP.NE.0.0) IPLTOP=6
     45               IF(NYR .GT. IY) GO TO 32
     46               IF (DATOPT .NE. 1) DATOPT = 0
     47        C     IBLK=BLOCK # OF DATA TO FOLLOW
     48        20     CALL SBLNK
     49               READ(5,111) IBLK
     50               IF(IBLK.EQ.0) GO TO 999
     51               NUM=0
     52               IF(IBLK.GT.IS17.AND.IBLK.LT.IC1.OR.IBLK.GT.IC16) GO TO 1
     53               IF(IBLK.GT.100) GO TO 200
     54               GO TO (20,1,1,1,20,1,20,20,20,2,1,3,1,2,20,20,20),IBLK
     55        200    GO TO (20,1,1,1,20,1,2,20,20,2,3,1,2,20,20,2),IBLK-100
```

```
      3     NUM=NUM+1
      2     NUM=NUM+1
      1     NUM=NUM+1
            IF(IBLK.EQ.IS5.OR.IBLK.EQ.IC5) GO TO 20
            READ(5,500) (COSTIN(IBLK,II),II=1,NUM)
500   FORMAT(1)
            GO TO 20
20    CONTINUE
C*** SET INDEX FOR READING CARD ON F10.0
            N=1
            M=10
C*** LOOP FOR ICROS NUMBER OF CARDS IN THIS BLOCK
            MX=((NYR-1)/10)+1
            DO 30 I=1,MX
C*** READ IN CARD
             READ (5,112) (COST(J),J=N,M)
112   FORMAT(10F8.2)
C*** CHANGE INDEX FOR NEXT CARD
              N=M+1
             M=M+10
30    CONTINUE
C*** WRITE DATA ON DIRECT ACCESS
            ITWO=L1
            IA=L2
            READ(8'IBLK,ERR=600) ISUB,CODE,TITLE
            WRITE(8'IBLK,ERR=600) ISUB,CODE,TITLE,COST
C   READ NEXT BLOCK .
            GOTO 20
   32 WRITE(6,34)
            STOP
   34 FORMAT(1H1,'NUMBER OF YEARS OF ANALYSIS EXCEEDS THE PARAMETER',
     +' IY. UPDATE IY OR CHANGE THE NUMBER OF YEARS FOR ANALYSIS.',/,
     +'THIS RUN WAS SELF TERMINATED.',/)
 111  FORMAT()
600   WRITE(6,601)
601   FORMAT(5X,'ERROR IN CSTDIN')
999   CONTINUE
            RETURN
C      DEBUG SUBTRACE,INIT(COST)
C      AT 100
C      TRACE ON
            END
```

```
APRT,S A.CSTDTO
```

SH*COST(1).CSTOTO

```
      1              SUBROUTINE CSTOTO
      2              INCLUDE PARS
      3              COMMON /NFLT/ FLATE
      4              COMMON/IP1/IP
      5              DIMENSION NIN(17)
      6              DATA NIN/2,2,2,2,2,2,3,3,4,4,4,4,4,4,3,4,4/
      7        C   READS HIGHEST LEVEL AND TRACE THROUGH EACH BRANCH TO COST DATA
      8        C**********************************
      9        C***  IF THIS ONLY AN ENERGY ANALYSIS COST DATA
     10        C***  FOR THE ENERGY LEVELS IS PRINTED
     11        C**********************************
     12              IF(INRGY .NE. 1) GO TO 40
     13              IRC1=IS1
     14              CALL PAGES(99)
     15              CALL PAGES(2)
     16              WRITE(6,1008)
     17        1008  FORMAT(1X,132('*'))
     18              READ(8'IS1,ERR=700) ISUB,CODE,TITLE,COST
     19              CALL PAGES(1)
     20              WRITE(6,1007) CODE,TITLE
     21              IF((COST(1)+COST(2)+COST(3)).EQ.0) GO TO 15
     22              CALL PAGES(5)
     23              WRITE(6,1009) CODE,TITLE,(COST(L),L=1,3)
     24        15    DO 20 II=1,K1
     25              IF(ISUB(II) .LE. 0) GO TO 20
     26              KL=ISUB(I)
     27              IF(KL.GT.110) KL=KL+1
     28              NUM=NIN(KL)
     29              CALL PAGES(NUM)
     30              READ(8'ISUB(II),ERR=700) ISE,CODE,TITLE,COST
     31              I=II
     32              CALL IFPRNT
     33        20    CONTINUE
     34              IRC1=IC1
     35              READ(8'IC1,ERR=700) ISUB,CODE,TITLE,COST
     36              CALL PAGES(1)
     37              WRITE(6,1007) CODE,TITLE
     38        1007  FORMAT(1X,3X,A5,2X,6A4)
     39              IF((COST(1)+COST(2)+COST(3)).LE.0) GO TO 30
     40              CALL PAGES(5)
     41              WRITE(6,1009) CODE,TITLE,(COST(L),L=1,3)
     42        30    DO 35 II=1,K1
     43              IF(ISUB(II).LE.0) GO TO 35
     44              READ(8'ISUB(II),ERR=700) ISE,CODE,TITLE,COST
     45              I=II
     46              CALL IFPRNT
     47        35    CONTINUE
     48              RETURN
     49        700   WRITE(5,601)
     50        601   FORMAT(5X,'ERROR IN CSTOTO')
     51              RETURN
     52        1009  FORMAT(1X,3X,A5,2X,6A4,/,2X,'COST INPUT OF THE SYSTEM',//
     53             +2X,' PRESENT VALUE =',F12.2,/,
     54             +2X,' ACTUAL COST   =',F12.2,/,
     55             +2X,' ANNUAL COST   =',F12.2)
```

```
      40    IRC1=IS1
      C*****************
      C***  HERE IF RUN IS REGULAR ANALYSIS OF SYSTEMS
      C*********************
            FLAT=FLATE*100.0
            DSC=DISC*100.0
      C*** WRITE STANDARD DATA
            CALL PAGES(99)
            WRITE(6,115) NYR,IYRST,IYR(NYR),DSC,FLAT
      C*** WRITE SUBLEVELS
         50 READ(8'IRC1,ERR=700) ISUB,CODE,TITLE,COST
            CALL PAGES(99)
            CALL PAGES(4)
            WRITE(6,100)
            CALL PAGES(1)
            WRITE(6,101) CODE,TITLE
             DO 60 II=2,K1
      C***********      WRITE SUBLEVEL CODES AND TITLES**************
               IF(ISUB(II).LE.0) GO TO 60
               READ(8'ISUB(II),ERR=700) ISB,CODE,TITLE
            CALL PAGES(1)
               WRITE(6,103) CODE,TITLE
         60  CONTINUE
      C    NEXT LEVEL
            DO 600 I=2,K1
            IF(ISUB(I).LE.0) GO TO 600
            READ(8'ISUB(I),ERR=700) ISA,CODE,TITLE,COST
      C*** WRITE CODE TITLE FOR THIS LEVEL
      C*** DETERMINE TYPE OF DATA
            KL=ISUB(I)
            IF(KL.GT.110) KL=KL-1
            IF(KL.GT.100) KL=KL-100
            NUM=MIN(KL)+1
            IF(KL.EQ.5.OR.KL.EQ.14) NUM=NUM+NYR
             IF(ISA(2).LE.0 .OR. COST(1) .GT. 0) GO TO 550
            CALL PAGES(4)
            WRITE(6,102) CODE,TITLE
            IF(ISUB(I).EQ.IRC1+1.OR.ISUB(I).EQ.IRC1+2.OR.
           +ISUB(I).EQ.IRC1+3.OR.ISUB(I).EQ. IRC1+5 .OR. ISUB(I) .EQ. IRC1+15)
           +    GO TO 150
            IF(ISUB(I).EQ.IRC1+7.OR.
           +ISUB(I).EQ.IRC1+8.OR.ISUB(I).EQ.IRC1+9.OR.ISUB(I).EQ.IRC1+13
           + .OR. ISUB(I) .EQ. IRC1+14) GO TO 200
            IF(ISUB(I).EQ.IRC1+6) GO TO 300
            IF(ISUB(I).EQ.IRC1+4)GO TO 400
            GO TO 550
      C   ACQUISITIONS  BLDG MODS SPACE
      C*** SEARCH LOWER LEVELS FOR DATA
       150     DO 190 J=2,K1
               IF(ISA(J).LE.0) GO TO 190
               READ(8'ISA(J),ERR=700) ISB,CODE,TITLE,COST
            CALL PAGES(NUM+4)
                WRITE(6,103) CODE,TITLE
            IF(CLTPRM.EQ.ISA(J)) WRITE(6,152) XLOAD,PCT(IP)
            IF(CLTPRM.EQ.ISA(J)) CALL PAGES(2)
      152   FORMAT(52X,'ANNUAL LOAD =',F12.2,' GJ/YR',/
```

```
112            +,83X,'PERCENT LOAD COVERED =',F12.2,'%')
113                 IF(ISB(2).GT.0)GOTO 155
114                 CALL IFPRNT
115                 GO TO 190
116          155    DO 180 K=2,K1
117                   IF(ISB(K).LE.0) GO TO 180
118                   READ(8'ISB(K),ERR=700) ISC,CODE,TITLE,
119               +COST
120                 CALL PAGES(NUM)
121                      WRITE(6,104) CODE,TITLE
122                      CALL IFPRNT
123          180    CONTINUE
124          190   CONTINUE
125                DOWNPM=DWNPMT*100.0
126                CALL PAGES(1)
127                WRITE(6,1109)DOWNPM
128         1109   FORMAT(1X,2X,'DOWNPAYMENT WILL BE ',F10.2,'% OF TOTAL COST')
129                  GO TO 600
130         C    THESE HAVE YEARLY COSTS
131           200   DO 290 J=2,K1
132         C*** SEARCH LOWER LEVELS FOR DATA
133                    IF(ISA(J).LE.0) GO TO 290
134                    READ(8'ISA(J),ERR=700) ISB,CODE,TITLE,COST
135                 CALL PAGES(NUM+4)
136                    WRITE(6,103)CODE,TITLE
137                    IF(ISB(2).GT.5) GO TO 250
138                    CALL IFPRNT
139                   GO TO 290
140          250      DO 280 K=1,K1
141                     IF(ISB(K).LE.0)GOTO 280
142                    READ(8'ISB(K),ERR=700) ISC,CODE,TITLE,
143               +COST
144                 CALL PAGES(NUM)
145                      WRITE(6,104) CODE ,TITLE
146                      CALL IFPRNT
147          280      CONTINUE
148          290    CONTINUE
149                GO TO 600
150         C    OPERATIONS COST DATA
151           300   DO 390 J=2,K1
152         C*** READ LOWER LEVELS FOR ANNUAL FUEL COST AND ESCALATION FACTOR
153                    IF(ISA(J).LE.0) GO TO 390
154                    READ(8'ISA(J),ERR=700) ISB,CODE,TITLE,
155               +COST
156                    COST(2)=COST(2)*100.0
157                 CALL PAGES(NUM+4)
158                    WRITE(6,103) CODE,TITLE
159                    WRITE(6,119) COST(1),COST(2)
160          390    CONTINUE
161                  GO TO 600
162         400    CONTINUE
163                DO 520 J=2,3
164         C*** READ LOWER LEVELS BY 3 RECORDS AT A TIME TEST FOR DATA
165         C*** PRINT ONES WITH DATA ...USED WITH REPLACEMENTS AND REPAIR
166                 J1=J+1
167                 J2=J+2
```

```
      IF(ISA(J).LE.0) GO TO 520
     CALL PAGES(NUM+4)
      READ(8'ISA(J),ERR=700) ISE,CODE,TITLE,COST
      IF(ISA(J1).LE.0) GO TO 410
      READ(8'ISA(J1),ERR=700) ISE,ACODE,ATITLE,ACOST
      IF(ISA(J2).LE.0) GO TO 405
      READ(8'ISA(J2),ERR=700) ISE,BCODE,BTITLE,BCOST
      WRITE(6,120) CODE,(TITLE(KA),KA=1,5),ACODE,(ATITLE(KB),KB=1,5),
     +BCODE,(BTITLE(KC),KC=1,5),
     +(IYR(K),COST(K),IYR(K),ACOST(K),IYR(K),BCOST(K),K=1,NYR)
      GO TO 520
405   CONTINUE
      WRITE(6,122)CODE,(TITLE(KA),KA=1,5),ACODE,(ATITLE(KB),KB=1,5),
     +(IYR(K),COST(K),IYR(K),ACOST(K),K=1,NYR)
      GO TO 520
410   CONTINUE
      WRITE(6,124)CODE,(TITLE(KA),KA=1,5),(IYR(K),COST(K),K=1,NYR)
520   CONTINUE
      GO TO 600
550   CALL PAGES(NUM+3)
      WRITE(6,102) CODE,TITLE
        CALL IFPRNT
600   CONTINUE
      IF(IRC1.EQ.IC1)RETURN
      IRC1=IC1
      GO TO 50
  100 FORMAT(1X,132('*'),/,1X,40X,'LISTING OF COST DATA INPUT',///)
  101 FORMAT(1X,A5,1X,6A8)
  102 FORMAT(1X,///,2X,A5,1X,6A8)
  103 FORMAT(1X,4X,A5,1X,6A8)
  104 FORMAT(1X,6X,A5,1X,6A8)
  115 FORMAT(1X,132('*'),/,1X,2X,'THIS HEATING VENTILATION AND COOLING',
     +' ANALYSIS',/,
     +3X,'WILL COVER  ',I5,' YEARS BEGINNING WITH',I5,' AND',/, 3X,
     +'ENDING WITH',I5,/,3X,'THE DISCOUNT RATE TO BE APPLIED TO ',
     +'THE ANALYSIS IS ---',F12.2,'%',/,3X,'THE INFLATION FACTOR TO BE',
     +' APPLIED TO THE ANALYSIS IS-',F12.2,'%',/,1X,131('*'))
  119 FORMAT(1X,50X,'ANNUAL FUEL COST =',F12.2,/,50X,'ESCALATION ',
     +'FACTOR =',2X,F12.2,'%',/)
  120  FORMAT(1X,/,3(2X,A5,5A8),/,3(3X,'YEAR',10X,'COST',23X),///,
     +5(3(2X,I5,2X,F12.2,23X),/),///)
  122  FORMAT(1X,/,2(2X,A5,5A8),/,2(3X,'YEAR',10X,'COST',23X),///,
     +5(2(2X,I5,2X,F12.2,23X),/),///)
  124  FORMAT(1X,/,(2X,A5,5A8),/,3X,'YEAR',10X,'COST',///,
     +5(2X,I5,2X,F12.2,/))
      END
```

```
>PRT.S A.CYPHER
```

SH*COST(I).CYPHER

```
      1              SUBROUTINE CYPHER(START,STOP,STEP)
      2              INTEGER CARD(80),EX,HYPHEN,ZERO,NINE,PERIOD,SPACE,SLASH
      3              DATA EX,HYPHEN,ZERO,NINE/0133,055,060,072/
      4              DATA PERIOD,SPACE,SLASH/056,040,057/
      5        1     CONTINUE
      6              READ(5,10) CARD
      7       10     FORMAT(80R1)
      8              VAR=0.0
      9              IFLAG=0
     10              START=0.0
     11              STOP=0.0
     12              STEP=0.0
     13              ISIGN=1
     14              I=1
     15       11     IF(CARD(I).NE.SPACE) GO TO 12
     16              I=I+1
     17              IF(I.GT.80) GO TO 999
     18              GO TO 11
     19       12     CONTINUE
     20              IF(CARD(I).EQ.HYPHEN) ISIGN =-1
     21              IF (CARD(I).EQ.HYPHEN) I=I+1
     22              IF(CARD(I).EQ.HYPHEN) GO TO 12
     23              IFLAG=IFLAG+1
     24       13     IF(CARD(I).EQ.PERIOD) GO TO 14
     25              IF(CARD(I).EQ.SPACE) GO TO 17
     26              IF(.NOT.(CARD(I).GE.ZERO.AND.CARD(I).LE.NINE)) STOP 'INVALID CARD'
     27              VAR=VAR*10.0+FLOAT(CARD(I)-ZERO)
     28       17     I=I+1
     29              IF(I.GT.80) GO TO 999
     30              GO TO 13
     31       14     I=I+1
     32              IF(I.GT.80) GO TO 999
     33              ICNT=0
     34              IFLAG=IFLAG+1
     35       15     IF(CARD(I).EQ.HYPHEN.OR.CARD(I).EQ.SPACE.OR.CARD(I).EQ.EX) GOTO 16
     36              ICNT=ICNT-1
     37              IF(.NOT.(CARD(I).GE.ZERO.AND.CARD(I).LE.NINE)) STOP 'BAD CARD'
     38              VAR=VAR*10.0+FLOAT(CARD(I)-ZERO)
     39              I=I+1
     40              IF(I.GT.80) GO TO 999
     41              GO TO 15
     42       16     VAR=(VAR*10.0**ICNT)*ISIGN
     43              ISIGN=1
     44              IF(IFLAG.EQ.2) START=VAR
     45              IF(IFLAG.EQ.4) STOP=VAR
     46              IF(IFLAG.EQ.6) STEP=VAR
     47              IF(IFLAG.EQ.6) GO TO 999
     48              VAR=0.0
     49       17     IF(CARD(I).NE.SPACE) GO TO 18
     50              I=I+1
     51              IF(I.GT.80) GO TO 999
     52              GO TO 17
     53       18     IF(IFLAG.EQ.3.AND.CARD(I).NE.HYPHEN) GO TO 999
     54              IF(IFLAG.EQ.4.AND.CARD(I).NE.EX) STOP 'NO X'
     55              I=I+1
```

```
    56              IF(I.ST.AD) GO TO 999
    57              GO TO 11
    58        999   RETURN
    59        C      DEBUG TRACE,SUBTRACE,INIT(CARD,VAR,START,STOP,STEP)
    60        C      AT 1
    61        C      TRACE ON
    62              END

@PRT,S A.DEPINC
```

SH*COST(1).DEPINC

```
      1            SUBROUTINE DEPINC
      2            INCLUDE PARS
      3            COMMON/NFLT/FLATE
      4      1      CONTINUE
      5            CALL SBLNK
      6      C**********************
      7      C***  ALL COMPUTATIONS IN THIS ROUTINE ARE FOR COMMERCIAL
      8      C**** APPLICATIONS ONLY DEPRECIATION DEDUCTS
      9      C***  ADDED INCOME DUE TO THE SHVAC SYSTEM IN RENTAL
     10      C***  DEDUCTIONS FROM TAXABLE INCOME DUE TO OPERATION EXPENSES
     11      C***  AND MAINTENANCE EXPENSES
     12      C**********************
     13      C    COMPUTE DEPRECIATION DEDUCTIONS AND AFTER TAX INCOME
     14            IF(IRC1.EQ.0) GO TO 150
     15            READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
     16      C*** TEST FOR INCOME TAX RATE IS <= 0 ERROR
     17            IF(COST(4) .GT. 0.0) GO TO 90
     18      C     WRITE(6,310)
     19            GO TO 150
     20      C    COST(1 THRU NYR)=DEPRECIATION IN YEAR I COST(NYR+1)=TAX RATE
     21      C********* COMPUTE STRAIGHT LINE DEPRECIATION*********
     22       90    N=COST(3)
     23            D=(COST(1)-COST(2))/COST(3)
     24             COST(NYR+1)=COST(4)
     25            DO 95 J=1,NYR
     26            IF(J .LE.N) COST(J)=D
     27       95   IF(J.GT.N)COST(J)=0.0
     28            DO 100 I=1,NYR
     29      C*** TOTAL HOLDS PV BY YEAR , PV , AC
     30      C***  COMPUTE PRESENT VAL ACTUAL AND ANNUALIZED COSTS
     31            ACTUAL(I)=COST(I)*COST(NYR+1)
     32            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
     33                TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
     34       100      TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
     35              TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
     36            WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     37           +FLCOST,TOTAL,ACTUAL
     38      C    COMPUTE ADDEDINCOME
     39        150 IF(IRC2 .EQ. 0) GO TO 300
     40            CALL SBLNK
     41            READ(8'IRC2,ERR=600) ISUB,CODE,TITLE,COST,
     42           +FLCOST
     43            IF (FLATE .EQ. 0.0) GO TO 250
     44            DO 200 I=1,NYR
     45            ACTUAL(I)=FLCOST(I)*(1.0-COST(2))
     46            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
     47            TOTAL(I)=ACTUAL(I)/(1.0+DISC)**II
     48       200  TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
     49            TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
     50            GO TO 290
     51      C    COST(1)=HVAC ADDED INCOME  COST(2)=INCOME TAX RATE
     52       250  TOTAL(NYR+2)=(1.0-COST(2))*COST(1)
     53            TOTAL(NYR+1)=TOTAL(NYR+2)*NYR/ACFACT
     54            DO 260 J=1,NYR
     55            ACTUAL(J)=TOTAL(NYR+2)
```

```
     260    ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(J)
     290    WRITE(9'IRC2,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST,TOTAL,ACTUAL
C***  MAINTENANCE DEDUCTIONS FROM INCOME TAX COMMERCIAL ONLY
     300    IF(IRC3 .LE. 0) GO TO 500
            CALL S9LNK
            READ(8'IRC3,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST
            DO 350 J=1,NYR
            IF(FLATE .EQ. 0.0) ACTUAL(J)=COST(1)*COST(2)
            IF(FLATE.NE. 0.0) ACTUAL(J)=FLCOST(J)*COST(2)
            TOTAL(J)=ACTUAL(J)/(1.0+DISC)**J
            TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(J)
            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(J)
     350    CONTINUE
            TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
            WRITE(8'IRC3,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST,TOTAL,ACTUAL
     500    IF(IRC9 .LE. 0) GO TO 700
            CALL S9LNK
C***  ENERGY DEDUCTS DUE TO OPERATIONS EXPENSES ******
            READ(8'IRC9,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST
            DO 550 I=1,NYR
            II=I+JYDSC
            ACTUAL(I)=COST(1)*(1.0+COST(2))**II*COST(3)
            TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
            TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
     550    CONTINUE
            TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
            WRITE(8'IRC9,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST,TOTAL,ACTUAL
     700    RETURN
     600    WRITE(6,601)
     601    FORMAT(5X,'ERROR IN DEPINC')
            RETURN
C        DEBUG TRACE,SUBTRACE,INIT
C        AT 1
C         TRACE ON
            END

PRT,S A.DIVVY
```

SH*COST(1).DIVVY

```
      1              SUBROUTINE DIVVY
      2              COMMON/NFLT/ FLATE
      3              COMMON/FLAG/IFLG
      4              COMMON /IP1/ IP
      5              INCLUDE PARS
      6              COMMON/INPUTS/COSTIN(200,3)
      7        C     DSSTRT  -  DISCOUNT RATE START VALUE
      8        C     DSSTOP  -  DISCOUNT RATE STOP VALUE
      9        C     DSSTEP  -   DISCOUNT RATE STEP VALUE
     10        C     FTSTRT  -  INFLATION RATE START RATE
     11        C     FTSTOP  -  INFLATION RATE STOP VALUE
     12        C     FTSTEP  -  INFLATION RATE STEP VALUE
     13        C     DPSTRT  -  DOWNPAYMENT START VALUE
     14        C     DPSTOP  -  DOWNPAYMENT STOP VALUE
     15        C     DPSTEP  -  DOWNPAYMENT STEP VALUE
     16        C     DISC  -  DISCOUNT RATE
     17        C     FLATE  -  INFLATION RATE
     18        C     DWNPRT  -  DOWNPAYMENT
     19
     20        900       CONTINUE
     21              DO 100 I=1,200
     22              IF(COSTIN(I,1).EQ.0.) GO TO 100
     23              DO 101 J=1,3
     24        101   COST(J)=COSTIN(I,J)
     25              READ(8'I) ISUB,CODE,TITLE
     26              WRITE(8'I) ISUB,CODE,TITLE,COST,FLCOST,TOTAL,ACTUAL
     27        100   CONTINUE
     28              IFLG=0
     29              IF(IW.NE.0) GO TO 13
     30              DISC=DSSTRT
     31              FLATE=FTSTRT
     32              DWNPRT=DPSTRT
     33              TINCOM=TSTART
     34              PTAX=PTSTRT
     35              CALL GFL
     36              IP=1
     37        15    READ(8'CLTPRM) ISUB,CODE,TITLE
     38              COST(1)=PRICE(IP)
     39              WRITE(8'CLTPRM) ISUB,CODE,TITLE,COST
     40              IRC=IS1
     41              IRC1=IS2
     42              GO TO 5
     43        1     IRC=IC1
     44              IRC1=IC2
     45        5     READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
     46              TACO=0
     47              IF(ISUB(2).EQ.0) GO TO 10
     48              DO 20 I=2,NTR
     49              IF(ISUB(I).EQ.0) GO TO 25
     50              READ(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST
     51        20    TACO=TACO+COST(1)
     52        25    GO TO 30
     53        10    TACO=COST(1)
     54        30    READ(8'IRC,ERR=600) ISUB
     55              DO 35 I=2,NTR
```

```
      IF(ISUB(I).LE.0.OR.ISUB(I).GT.200) GO TO 40
      IF(ISUB(I).GE.IS2.AND.ISUB(I).LE.IS5) GO TO 35
      IF(ISUB(I).GE.IC2.AND.ISUB(I).LE.IC5) GO TO 35
      READ(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST
      IF(ISA(2).NE.0) GO TO 45
      CALL LOAD(TACO)
      WRITE(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST
      GO TO 35
45    DO 50 J=2,NTR
      IF(ISA(J).EQ.0) GO TO 35
      READ(8'ISA(J),ERR=600) IS9,CODE,TITLE,COST
      CALL LOAD(TACO)
50    WRITE(8'ISA(J),ERR=600) IS9,CODE,TITLE,COST
35    CONTINUE
40    IF(IRC.EQ.IS1) GO TO 1
      IW=99
      RETURN
13    IP=IP+1
      IF(IPLTOP.EQ.1.AND.PRICE(IP+1).EQ.0.OR.IP.GE.24) IFLG=1
      IF(PRICE(IP).NE.0.0.AND.IP.LE.25) GO TO 15
      IP=1
      DISC=DISC+DSSTEP
      IF(IPLTOP.EQ.2.AND.DISC+DSSTEP.GE.DSSTOP) IFLG=1
      IF(DISC.LE.DSSTOP.AND.DSSTEP.NE.0) GO TO 15
      DISC=DSSTRT
      PTAX=PTAX+PTSTEP
      IF(IPLTOP.EQ.5.AND.PTAX+PTSTEP.GE.PTSTOP) IFLG=1
      IF(PTAX.LE.PTSTOP.AND.PTSTEP.NE.0) GO TO 15
      PTAX=PTSTRT
      TINCOM=TINCOM+TSTEP
      IF(IPLTOP.EQ.6.AND.TINCOM+TSTEP.GE.TSTOP) IFLG=1
      IF(TINCOM.LE.TSTOP.AND.TSTEP.NE.0) GO TO 15
      TINCOM=TSTART
      FLATE=FLATE+FTSTEP
      IF(IPLTOP.EQ.3.AND.FLATE+FTSTEP.GE.FTSTOP) IFLG=1
      IF(FLATE.LE.FTSTOP.AND.FTSTEP.NE.0) GO TO 15
      FLATE=FTSTRT
      DWNPMT=DWNPMT+DPSTEP
      IF(IPLTOP.EQ.4.AND.DWNPMT+DPSTEP.GE.DPSTOP) IFLG=1
      IF(DWNPMT.LE.DPSTOP.AND.DPSTEP.NE.0) GO TO 15
      DISC=-999.0
      IFLG=1
600   RETURN
C          DEBUG INIT,SUBTRACE,TRACE
C          AT 900
C          TRACE ON
      END
```

SH*COST(1).ENERGY

```
      1            SUBROUTINE ENERGY
      2            INCLUDE PARS
      3            COMMON/NFLT/FLATE
      4      C   COMPUTATION OF ENERGY COSTS FOR HVAC
      5      C   COSTS FOR EACH TYPE OF FUEL OR ENERGY SOURCE PER YEAR ARE
      6      C   ESCALATED. EACH SUBLEVEL CONTAINS ANNUAL ENERGY COST AND THE
      7      C   CORRESPONDING ESCALATION FACTOR. TOTALS ARE PLACED IN OPERATIONS
      8      C   LEVEL IN TOTAL
      9      1     CONTINUE
     10            IF(IRC1.EQ.0)GO TO 110
     11            CALL SBLNK
     12            READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     13           +AFCOST
     14      C*** IF THIS LEVEL IS EMPTY GO ON TO NEXT
     15            IF(ISUB(2) .GT. 0) GO TO 40
     16      C*** CHECK FOR POSITIVE COSTS
     17            IF(COST(1) .LE. 0.0) GO TO 110
     18            DO 25 I=1,NYR
     19            II=I+JYDSC
     20             ACTUAL(I)=COST(1)*(1.0+COST(2))**II
     21            TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
     22            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
     23      25    TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
     24            GO TO 105
     25      C*** HERE IF NO SUBLEVELS
     26      C*** COMPUTE COSTS IN TERMS OF PRESENT VALUE
     27      C     READ EACH FUEL TYPE AND COMPUTE YEARLY COSTS
     28        40  DO 100 I=2,K1
     29      C*** SUBLEVEL COSTS
     30              IF(ISUB(I).LE.0)GOTO 100
     31            ISUBI=ISUB(I)
     32              READ(8'ISUBI,ERR=610) ISA,ACODE,ATITLE,
     33           +ACOST,FLCOST
     34      C    COST(1)=FUEL COST , COST(2)= ESCALATION FACTOR
     35            DO 65 J=1,NYR
     36            JJ=J+JYDSC
     37            BCTUAL(J)=ACOST(1)*((1.0+ACOST(2))**JJ)
     38            ATOTAL(J)=BCTUAL(J)/(1.0+DISC)**J
     39            TOTAL(J)=TOTAL(J)+ATOTAL(J)
     40            ACTUAL(J)=ACTUAL(J)+BCTUAL(J)
     41            ATOTAL(NYR+1)=ATOTAL(NYR+1)+ATOTAL(J)
     42            BCTUAL(NYR+1)=BCTUAL(NYR+1)+BCTUAL(J)
     43      65    CONTINUE
     44            ATOTAL(NYR+2)=ATOTAL(NYR+1)*ACFACT
     45      C   ATOTAL ARE FOR THIS SUBLEVEL
     46      C*** ADD SUBLEVEL TOTALS TO UPPER LEVEL TOTALS
     47         90     WRITE(8'ISUB(I),ERR=600) ISA,ACODE,ATITLE,ACOST,
     48           +FLCOST,ATOTAL,BCTUAL
     49              TOTAL(NYR+1)=TOTAL(NYR+1)+ATOTAL(NYR+1)
     50            ACTUAL(NYR+1)=ACTUAL(NYR+1)+BCTUAL(NYR+1)
     51              ATOTAL(NYR+1)=0.0
     52            BCTUAL(NYR+1)=0.0
     53       100    ATOTAL(NYR+2)=0.0
     54      C   TOTAL ARE FOR THE OPERATIONS LEVEL
     55       105 TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
```

```
    56              WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
    57             +AFCOST,TOTAL,ACTUAL
    58           110 RETURN
    59         600   WRITE(6,601)
    60         601   FORMAT(5X,'ERROR IN ENERGY')
    61               RETURN
    62         C      DEBUG INIT,SUBTRACE,TRACE
    63         C      AT 1
    64         C      TRACE ON
    65               END
```

```
@PRT,S A.GFL
```

SH*COST(1).GFL

```
      1              SUBROUTINE GFL
      2                  COMMON/NAM/NAMES
      3              COMMON/LAST/PRICEL
      4              COMMON/GFL2/SSTART,SSTOP,SSTEP
      5              INCLUDE PARS
      6              REAL LD,L
      7              DIMENSION FRTAO(2),FRTA(2),FRUL(2)
      8              CHARACTER*63 NAMES
      9              DATA FRTAO,FRTA,FRUL/.50,.75,.60,.75,4.00,4.00/
     10        1         CONTINUE
     11              I=LCN
     12              J=LOA
     13              L=XLOAD
     14
     15              CALL TABLE(I,J,PA,PB,PC,PD,PE,PF,NAMES,LO)
     16              K=0
     17              AC=SSTART
     18        1000  K=K+1
     19              PRICE(K)=AC*CSTPSM+0.0005
     20              AREA(K)=AC
     21              X=(FRUL(J)/FRTA(J))-8.0
     22              Y=(FRTA(J)/FRTA0(J))*(LD/L)*AC
     23              R=PA+PB*X+PC*(X**2)
     24              S=PD+PE*X+PF*(X**2)
     25              EXPNT=(R*Y+S*(Y**2))*(-1)
     26              FPCT=1-EXP(EXPNT)
     27              AC=AC+SSTEP
     28              G(K)=FPCT*L
     29              PCT(K)=FPCT*100.0
     30              IF(AC.LE.SSTOP.AND.K.LT.IGFL)GO TO 1000
     31              PRICEL=PRICE(K)
     32              RETURN
     33        C      DEBUG INIT,TRACE,SUBTRACE
     34        C      AT 1
     35        C      TRACE ON
     36              END
```

@PRT,S A.IFPRNT

SH*COST(1).IFPRNT

```
      1              SUBROUTINE IFPRNT
      2              INCLUDE PARS
      3        C   DETERMINE FORMAT AND VARIABLES FOR PRINT
      4              IF(ISUB(I).EQ.0) GO TO 100
      5              COST(NYR+1)=COST(NYR+1)*100
      6              COST(NYR+2)=COST(NYR+2)*100
      7              ISWTCH=ISUB(I)-1
      8              IF(ISWTCH .GT. 100) GO TO 1
      9              GO TO (10,15,15,20,25,30,35,40,45,50,55,60,65,69,85,87),ISWTCH
     10        1      GO TO (10,15,15,20,25,30,35,40,45,55,60,65,69,85,87),ISWTCH-100
     11        C   ACQUISITIONS
     12        10     CONTINUE
     13               WRITE(6,70) COST(1)
     14               GO TO 100
     15        C    BUILDING MODIFICATIONS
     16        C      SPACE OCCUPIED
     17        15     CONTINUE
     18               WRITE(6,71)COST(1)
     19               GO TO 100
     20        C   NET REPAIRS AND REPLACEMENTS
     21        20     CONTINUE
     22               WRITE(6,73)(IYR(K),COST(K),K=1,NYR)
     23               GO TO 100
     24        C      MAINTENANCE COSTS
     25        25     CONTINUE
     26               WRITE(6,72) COST(1)
     27               GO TO 100
     28        C    OPERATIONS COSTS
     29        30     COST(2)=COST(2)*100.0
     30               WRITE(6,75) COST(1),COST(2)
     31               GO TO 100
     32        C   PROPERTY TAX COSTS
     33        35     COST(2)=COST(2)*100.0
     34               WRITE(6,76) COST(2),COST(1)
     35               GO TO 100
     36        C   PROPERTY TAX DEDUCTIONS
     37        40     COST(2)=COST(2)*100.0
     38               COST(3)=COST(3)*100.0
     39               WRITE(6,77) COST(2),COST(3),COST(1)
     40               GO TO 100
     41        C    DEPRECIATION DEDUCTIONS
     42        45     COST(4)=COST(4)*100.0
     43                WRITE(6,78) (COST(K),K=1,4)
     44               GO TO 100
     45        C    ADDITIONAL AFTER TAX INCOME
     46        50     COST(2)=COST(2)*100.0
     47               WRITE(6,79)COST(1),COST(2)
     48               GO TO 100
     49        C    INSURANCE COSTS
     50        55     CONTINUE
     51               WRITE(6,80) COST(1),COST(2),COST(3)
     52               GO TO 100
     53        C    SALVAGE VALUE AFTER N YEARS OF ANALYSIS
     54        60     CONTINUE
     55               WRITE(6,81) COST(1),NYR
```

```
      GO TO 100
C   LOAN MORTGAGE VALUE
65    CONTINUE
      WRITE(6,82) COST(NYR+1),COST(NYR+2),
     +(IYR(K),COST(K),K=1,NYR)
      GO TO 100
69    CONTINUE
      WRITE(6,83)COST(NYR+1),COST(1)
      GO TO 100
85    COST(2)=COST(2)*100.0
      WRITE(6,84)COST(1),COST(2)
      GO TO 100
87    COST(2)=COST(2)*100.0
      COST(3)=COST(3)*100.0
      WRITE(6,83)COST(1),COST(2),COST(3)
 70 FORMAT(1X,50X,'INITIAL COST =',F12.2,/)
 71 FORMAT(1X,50X,'COST =',F12.2,/)
 72 FORMAT(1X,50X,' COST PER YEAR = ',F12.2,/)
 73 FORMAT(1X,50X,'YEAR',10X,'COST',/,50(50X,I5,2X,F12.2,/))
 75 FORMAT(1X,50X,'ANNUAL FUEL COST =',F12.2,/,50X,'ESCALATION ',
     +'FACTOR =',2X,F10.2,'%',/)
 76 FORMAT(1X,50X,'PROPERTY TAX RATE =',F10.2,'%',/,51X,
     +'ASSESSED VALUE     =',F10.2)
 77 FORMAT(1X,50X,'PROPERTY TAX RATE =',F10.2,'%',/,50X,
     +1X,'INCOME TAX RATE   =',F10.2,'%',/,51X,'ASSESSED',
     +' VALUE     =',F10.2)
 78    FORMAT(1X,50X,'ORIGINAL VALUE',7X,'=',1X,F12.2,/,51X,'SALVAGE ',
     +'VALUE',9X,'=',F12.2,/,51X,'YEARS OF EXPECTED USE =',F12.2,/,51X,
     +'INCOME TAX RATE       =',3X,F10.2,'%',/)
 79 FORMAT(1X,50X,'ADDITIONAL ANNUAL INCOME =',F12.2,/,51X,'INCOME ',
     +'TAX RATE =',11X,F10.2,'%',/)
 80   FORMAT(1X,50X,'ANNUAL PREMIUMS =',16X,F12.2,/,51X,
     +'ANNUAL ESTIMATED DAMAGE =',8X,F12.2,/,51X,
     +'ANNUAL INSURANCE REIMBURSEMENTS =',F12.2,/)
 81 FORMAT(1X,50X,'SALVAGE VALUE =',F12.2,' AFTER',I5,' YEARS',/)
 82 FORMAT(1X,50X,'MARKET INTEREST RATE =',F10.2,'%',/,51X,
     +'INCOME TAX RATE =',5X,F10.2,'%',/,51X,'YEAR',3X,'OUTSTANDING ',
     +'MORTGAGE VALUE',/,50(50X,I5,6X,F12.2,/))
 83    FORMAT(1X,50X,'MARKET INTEREST RATE =',F10.2,'%',/,51X,
     +'ANNUAL PAYMENT  =',5X,F10.2)
 84    FORMAT(1X,50X,' COST PER YEAR = ',F12.2,/,51X
     +,'INCOME TAX RATE =',2X,F10.2,'%',/)
 89    FORMAT(1X,50X,'ANNUAL FUEL COST =',1X,F12.2,/,51X,'ESCALATION',
     +' FACTOR =  ',F10.2,'%',/,51X,'INCOME TAX RATE =',4X,F10.2,'%',/)
100   100  RETURN
      END
```

$PRT,S A,LEGEND

SH*COST(1).LEGEND

```
      1              SUBROUTINE LEGEND(LABELS,IEX,ICODE)
      2              CHARACTER*8 LABELS(18)
      3              CHARACTER*2 EXP
      4                  IF(ICODE.NE.1) GO TO 14
      5              CALL SYMBOL(3.5,9.0,.143,6HLEGEND,0.0,6)
      6              YAXIS=8.0
      7              I=0
      8                  XNUM=0
      9              YSTEP=0.2
     10        14       NNUM=0
     11              DO 11 J=1,8
     12              IF(LABELS(J).EQ.8H        ) GO TO 12
     13        11    NNUM=NNUM+8
     14        12    CONTINUE
     15              YAX=YAXIS-(YSTEP*FLOAT(I))
     16              I=I+1
     17              CALL SYMBOL(0.0,YAX,.143,LABELS,0.0,NNUM)
     18              XAX=FLOAT(NNUM)/7.0
     19              CALL SYMBOL(XAX,YAX,.143,8H X10,0.0,8)
     20              ENCODE(2,13,EXP) IEX
     21        13    FORMAT(I2)
     22              CALL SYMBOL(XAX+.6,YAX+.125,.072,EXP,0.0,2)
     23              XNUM=XNUM+1
     24              XAXIS=6.0
     25              CALL NUMBER(XAXIS,YAX,.071,XNUM,0.0,-1)
     26              DO 10 J=1,2
     27              XAXIS=6.0+FLOAT(J)
     28              CALL PLOT(XAXIS,YAX,2)
     29        10    CALL NUMBER(XAXIS,YAX,.071,XNUM,0.0,-1)
     30              IF(ICODE.EQ.3) CALL PLOT(15.0,0.0,-3)
     31              RETURN
     32              END
```

$PRT,S A.LIFCYC

SH•COST(1).LIFCYC

```
      1            SUBROUTINE LIFCYC
      2            INCLUDE PARS
      3            COMMON/CYLIF/CYLF(6)
      4            EQUIVALENCE (CYLIFE,CYLF(1)),(ACYLIF,CYLF(2)),(ANCYLF,CYLF(3)),
      5           •(FUSAVE,CYLF(4)),(AFUSAV,CYLF(5)),(ANFUSV,CYLF(6))
      6      C•••••••••••••••••••••••
      7      C•••       READ RECORDS FOR TOTAL COSTS OF EACH SYSTEM
      8      C•••       IS1 IS TOP LEVEL FOR SHVAC
      9      C•••       IC1 IS TOP LEVEL FOR CHVAC
     10      C•••••••••••••••••••••••
     11            IF(IS1 .NE. 0) READ(8'IS1,ERR=600) ISE,ECODE,ETITLE,COST,ETOTAL,
     12           •TOTAL,ACTUAL,PV,AC,ACT
     13            IF(IC1 .NE. 0) READ(8'IC1,ERR=600)ISE,ECODE,ETITLE,ACOST,ETOTAL,
     14           •ATOTAL,BCTUAL,PV1,AC1,ACT1
     15      C•••••••••••••••••••••••
     16      C•••       READ ENERGY COSTS FOR SHVAC (IS7) AND FOR CHVAC (IC7)
     17      C•••••••••••••••••••••••
     18            READ(8'IS7,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,BTOTAL,
     19           •CCTUAL
     20            READ(8'IC7,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,CTOTAL,
     21           •DCTUAL
     22      C••••••••••••••••••••••••••
     23      C•••       COMPUTE LIF CYCLE COSTS AND FUEL SAVINGS
     24      C••••••••••••••••••••••••••
     25            CYLIFE=PV1-PV
     26            ACYLIF=ACT1-ACT
     27            ANCYLF=AC1-AC
     28      C••••••••••••••••••••••••••
     29      C•••       IF ANALYSIS IS COMMERCIAL ENERGY DEDUCTS MUST BE DEDUCTED
     30      C•••       ELSE ONLY ENERGY COSTS ARE INVOLVED
     31      C••••••••••••••••••••••••••
     32                 IF(ICOM .EQ. 1) GO TO 100
     33            FUSAVE=CTOTAL(NYR+1)-BTOTAL(NYR+1)
     34            AFUSAV=DCTUAL(NYR+1)-CCTUAL(NYR+1)
     35            ANFUSV=CTOTAL(NYR+2)-BTOTAL(NYR+2)
     36      C•••••••••••••••••••••••
     37      C•••       SKIP COMMERCIAL ANALYSIS IF HERE GO TO CHECK FOR
     38      C•••       ENERGY ANALYSIS ONLY OR REGULAR TYPE
     39      C•••••••••••••••••••••••
     40                 GO TO 200
     41      C•••••••••••••••••••••
     42      C•••       READ ENERGY COSTS AND ENERGY DEDUCTS FROM IS7 AND IS17
     43      C•••       DETERMINE NET FUEL COSTS FOR FUEL SAVINGS
     44      C•••••••••••••••••••••
     45      100   READ(8'IS7,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,BTOTAL,
     46           •CCTUAL
     47            READ(8'IS17,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,CTOTAL,
     48           •DCTUAL
     49                 FUNET1=BTOTAL(NYR+1)-CTOTAL(NYR+1)
     50                 AFNET1=CCTUAL(NYR+1)-DCTUAL(NYR+1)
     51                 ANFNT1=BTOTAL(NYR+2)-CTOTAL(NYR+2)
     52      C•••••••••••••••••••••
     53      C•••       READ CHVAC ENERGY COSTS AND ENERGY DEDUCTS
     54      C•••       DETERMINE FUEL SAVINGS
     55      C•••••••••••••••••••••
```

```
     READ(8'IC7,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,BTOTAL,
    +CCTUAL
     READ(8'IC16,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,CTOTAL,
    +DCTUAL
          FUNET2=BTOTAL(NYR+1)-CTOTAL(NYR+1)
          AFNET2=CCTUAL(NYR+1)-DCTUAL(NYR+1)
          ANFNT2=BTOTAL(NYR+2)-CTOTAL(NYR+2)
          FUSAVE=FUNET2-FUNET1
          AFUSAV=AFNET2-AFNET1
          ANFUSV=ANFNT2-ANFNT1
C*********************
C***      IF THIS IS AN ENERGY COSTS ANALYSIS ONLY
C***      RECOMPUTE LIFE CYCLE COSTS
C***      ELSE GO WRITE RESULTS TO FILE 8
C*********************
200   IF(INRGY .NE. 1) GO TO 300
      CYLIFE=FUSAVE-(COST(1)-ACOST(1))
      ANCYLF=ANFUSV-(COST(2)-ACOST(2))
      ACYLIF=AFUSAV-(COST(3)-ACOST(3))
C*********************
C***      WRITE THE LIFE CYCLE COSTS OUT TO MASS STORAGE
C*********************
300   READ(8'ILF,ERR=600) ISUB,CODE,TITLE
      WRITE(8'ILF,ERR=600) ISUB,CODE,TITLE,CYLF
           RETURN
600        WRITE(6,601)
601        FORMAT(5X,'ERROR IN LIFCYC')
           RETURN
      END
```

```
$PRT,S A.LOAD
```

SH*COST(1).LOAD

```
      1             SUBROUTINE LOAD(TACO)
      2             INCLUDE PARS
      3             COMMON/IP1/IP
      4             IBLK=ISUB(I)
      5             IF(IBLK.GT.100) GO TO 100
      6             GO TO (9,9,9,9,9,10,9,20,25,30,65,40,9,50,45,55,9),IBLK
      7       100   GO TO (9,9,9,9,9,10,15,23,25,30,40,9,50,75,70,60),IBLK-100
      8       55    READ(8'IS6,ERR=600) ISD,DCODE,DTITLE,COST
      9             COST(2)=TINCOM
     10             RETURN
     11       70    READ(8'IC6,ERR=600) ISD,DCODE,DTITLE,COST
     12             COST(2)=TINCOM
     13             RETURN
     14       10    COST(1)=COST(1)*TACO
     15             RETURN
     16       25    COST(3)=TINCOM
     17       20    COST(1)=TACO
     18             COST(2)=PTAX
     19             RETURN
     20       30    COST(4)=TINCOM
     21             COST(3)=COST(2)
     22             COST(2)=COST(1)
     23             COST(1)=TACO
     24             RETURN
     25       40    COST(1)=COST(1)*TACO
     26             RETURN
     27       65    COST(2)=TINCOM
     28             RETURN
     29       45    READ(8'IS4,ERR=600) ISD,DCODE,DTITLE,COST
     30             RETURN
     31       75    READ(8'113,ERR=600) ISD,DCODE,DTITLE,COST
     32             RETURN
     33       50    COST(4)=TINCOM
     34             COST(3)=COST(2)
     35             COST(2)=COST(1)
     36             COST(1)=TACO*(1.-DWNPRT)
     37             RETURN
     38       60    READ(8'IC7,ERR=600) ISD,DCODE,DTITLE,COST
     39             COST(3)=TINCOM
     40             ACOST(1)=COST(1)*(1.-PCT(IP)/100.)
     41             ACOST(2)=COST(2)
     42             ACOST(3)=COST(3)
     43             READ(8'IC7,ERR=600) ISD,DCODE,DTITLE
     44             WRITE(8'IC7,ERR=600) ISD,DCODE,DTITLE,ACOST
     45             RETURN
     46       15    ACOST(1)=COST(1)*(1.-PCT(IP)/100.)
     47             ACOST(2)=COST(2)
     48             READ(8'IS7,ERR=600) ISD,DCODE,DTITLE
     49             WRITE(8'IS7,ERR=600) ISD,DCODE,DTITLE,ACOST
     50             RETURN
     51       600   PRINT *,'ERROR IN LOAD'
     52       9     RETURN
     53             END
```

```
@PRT.S A.RETREP
```

SH*COST(L).NETREP

```
      1            SUBROUTINE NETREP
      2            INCLUDE PARS
      3            COMMON/NFLT/FLATE
      4      C   COMPUTES NET REPLACEMENT COSTS AND MAINTENANCE COSTS
      5      C***  TESTS FOR INFLATED OR REGULAR COSTS AND USES THE APPROPRIATE ONE
      6      C***  COMPUTES PRESENT VAL ACTUAL AND ANNUALIZED COSTS
      7            IF(IRC1.EQ. 0) GO TO 210
      8            CALL SBLNK
      9            READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     10           *FLCOST
     11            IF(ISUB(2).GT. 0)GO TO 85
     12            CHK=0.0
     13            DO 30 I=1,NYR
     14     30     CHK=CHK+COST(I)
     15            IF(CHK .LE. 0.0) GO TO 210
     16            DO 70 I=1,NYR
     17     C***   CHECK FOR INFLATION IF TRUE USE FLCOST ELSE USE COST
     18            IF(FLATE.NE.0.0)ACTUAL(I)=FLCOST(I)
     19            IF(FLATE.EQ.0.0)ACTUAL(I)=COST(I)
     20            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
     21            TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
     22     70     TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
     23            GO TO 205
     24     C     TOTAL(1<>NYR) ARE COST FOR YEAR I
     25     C    TOTAL(NYR+1) IS PRESENT VALUE COST
     26     C    TOTAL(NYR+2) IS ANNUAL COST)
     27      85     DO 200 II=2,K1
     28     C**************
     29     C***   IF HERE THERE ARE SUBLEVELS TO BE COMPUTED
     30            IF(ISUB(II).LE.0) GO TO 200
     31            READ(8'ISUB(II),ERR=600) ISA,ACODE,ATITLE,ACOST,
     32           *AFCOST
     33            DO 90 I=1,NYR
     34            IF(FLATE.NE.0.0)BCTUAL(I)=AFCOST(I)
     35            IF(FLATE.EQ.0.0)BCTUAL(I)=ACOST(I)
     36            BCTUAL(NYR+1)=BCTUAL(NYR+1)+BCTUAL(I)
     37             ATOTAL(I)=BCTUAL(I)/(1.0+DISC)**I
     38            ACTUAL(I)=BCTUAL(I)+ACTUAL(I)
     39            TOTAL(I)=TOTAL(I)+ATOTAL(I)
     40     90     ATOTAL(NYR+1)=ATOTAL(NYR+1)+ATOTAL(I)
     41            ATOTAL(NYR+2)=ATOTAL(NYR+1)+ACFACT
     42            WRITE(8'ISUB(II),ERR=600) ISA,ACODE,ATITLE,ACOST,
     43           *AFCOST,ATOTAL,BCTUAL
     44            ACTUAL(NYR+1)=ACTUAL(NYR+1)+BCTUAL(NYR+1)
     45            TOTAL(NYR+1)=TOTAL(NYR+1)+ATOTAL(NYR+1)
     46            ATOTAL(NYR+1)=0.0
     47            ATOTAL(NYR+2)=0.0
     48            BCTUAL(NYR+1)=0.0
     49     200    CONTINUE
     50     205    TOTAL(NYR+2)=TOTAL(NYR+1)+ACFACT
     51            WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     52           *FLCOST,TOTAL,ACTUAL
     53     210    IF(IRC2 .EQ. 0) GO TO 550
     54            CALL SBLNK
     55     C    MAINTENANCE COSTS
```


C.2

```
     56              READ(8'IRC2,ERR=600) ISUB,CODE,TITLE,COST,
     57             +FLCOST
     58              IF(ISUB(2) .GT. 0) GO TO 300
     59              DO 240 J=1,NYR
     60              IF(FLATE.NE.0.0)ACTUAL(J)=FLCOST(J)
     61              IF(FLATE.EQ.0.0)ACTUAL(J)=COST(1)
     62              TOTAL(J)=FLCOST(J)/(1.0+DISC)**J
     63              ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(J)
     64       240    TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(J)
     65              TOTAL(NYR+2)=TOTAL(NYR+1)+ACFACT
     66              GO TO 500
     67       300    DO 350 J=2,K1
     68              IF(ISUB(J) .LE.0)GO TO 350
     69              READ(8'ISUB(J),ERR=600) ISA,ACODE,ATITLE,ACOST,
     70             +AFCOST
     71       C***   IF HERE SUBLEVELS ARE TO BE COMPUTED
     72              DO 320 J1=1,NYR
     73              IF(FLATE .EQ. 0) BCTUAL(J1)=COST(1)
     74              IF(FLATE .NE. 0)BCTUAL(J1)=AFCOST(J1)
     75              BCTUAL(NYR+1)=BCTUAL(J1)+BCTUAL(NYR+1)
     76              ACTUAL(J1)=ACTUAL(J1)+BCTUAL(J1)
     77              ATOTAL(J1)=BCTUAL(J1)/(1.0+DISC)**J1
     78              TOTAL(J1)=TOTAL(J1)+ATOTAL(J1)
     79       320    ATOTAL(NYR+1)=ATOTAL(NYR+1)+ATOTAL(J1)
     80              ATOTAL(NYR+2)=ATOTAL(NYR+1)+ACFACT
     81              TOTAL(NYR+1)=TOTAL(NYR+1)+ATOTAL(NYR+1)
     82              ACTUAL(NYR+1)=ACTUAL(NYR+1)+BCTUAL(NYR+1)
     83              WRITE(8'ISUB(J),ERR=600) ISA,ACODE,ATITLE,ACOST,
     84             +AFCOST,ATOTAL,BCTUAL
     85              ATOTAL(NYR+1)=0.0
     86              ATOTAL(NYR+2)=0.0
     87              BCTUAL(NYR+1)=0.0
     88       350    CONTINUE
     89              TOTAL(NYR+2)=TOTAL(NYR+1)+ACFACT
     90       500    WRITE(8'IRC2,ERR=600) ISUB,CODE,TITLE,COST,
     91             +FLCOST,TOTAL,ACTUAL
     92       550    RETURN
     93       600    WRITE(6,601)
     94       601    FORMAT(5X,'ERROR IN NETREP')
     95              RETURN
     96              END

$PRT,S A.NFLATE
```

SH*COST(1),NFLATE

```
      1            SUBROUTINE NFLATE
      2            INCLUDE PARS
      3                COMMON /NFLT/ FLATE
      4      1      CONTINUE
      5            CHK=0
      6            I=2
      7      C*****************************
      8      C***  READ TOP LEVEL IF  NO COSTS ARE HERE SEARCH LOWER LEVELS
      9      C*****************************
     10            READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
     11      1001  FORMAT(1X,3X,'ISUBS>>',20(2X,I4))
     12            FLAT=FLATE+1.0
     13            IF(COST(1) .EQ. 0) GO TO 50
     14            DO 40 I=1,NYR
     15            II=I+JYDSC
     16      40    FLCOST(I)=COST(I)*FLAT**II
     17            WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     18           *FLCOST
     19            GO TO 520
     20      C*******************
     21      C***  DETERMINE TYPE OF DATA AND DECIDE IF INFLATION APPLIES
     22      C*************************
     23      50    ISWTCH=ISUB(I)-IRC1
     24            IF(ISWTCH .GT.16 .OR. ISWTCH .LE. 0) GO TO 300
     25            READ(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST
     26            IF(ISUB(I).GT.100) GO TO 100
     27            GO TO (350,350,350,55,250,350,250,250,350,250,350,350,350,350,
     28           *250,350,350),ISWTCH
     29      100   GO TO (350,350,350,55,250,350,250,250,350,350,350,350,350,
     30           *250,350,350),ISWTCH
     31      C***************
     32      C***  CHECK COSTS IN LEVELS UNTIL NOT = ZERO
     33      C***  WHEN NOT 0 INFLATE COSTS AND PUT IN FLCOST
     34      C************************
     35      55    DO 60 J=1,NYR
     36      60    CHK=CHK+COST(J)
     37            IF (CHK .NE. 0.0)GO TO 200
     38            I1=2
     39      65    IF(ISA(I1) .LE. 0)GO TO 165
     40            READ(8'ISA(I1),ERR=600) ISB,CODE,TITLE,COST
     41            DO 70 J=1,NYR,
     42      70    CHK=CHK+COST(J)
     43            IF (CHK .NE. 0.0) GO TO 150
     44                I2=2
     45      75    IF(ISB(I2) .LE. 0) GO TO 110
     46                READ(8'ISB(I2),ERR=600) ISC,CODE,TITLE,COST
     47                DO 80 J=1,NYR
     48      80        CHK=CHK+COST(J)
     49                IF(CHK .LE. 0.0 )GO TO 300
     50                DO 90 J=1,NYR
     51            JJ=J+JYDSC
     52      90        FLCOST(J)=COST(J)*FLAT**JJ
     53                WRITE(8'ISB(I2),ERR=600) ISC,CODE,TITLE,COST,
     54           *FLCOST
     55      110   I2=I2+1
```

```
      CHK=0.0
           IF (I2 .LE.K1) GO TO 75
           GO TO 165
  150      DO 160 J=1,NYR
      JJ=J+JYDSC
  160      FLCOST(J)=COST(J)*FLAT**JJ
        WRITE(8'ISA(I1),ERR=600) ISB,CODE,TITLE,COST,
     +FLCOST
  165      I1=I1+1
      CHK=0.0
           IF(I1 .LE.K1) GO TO 65
       GO TO 240
  200      DO 210 J=1,NYR
      JJ=J+JYDSC
  210      FLCOST(J)=COST(J)*FLAT**JJ
           WRITE(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST,
     +FLCOST
  240   I=I+1
      CHK=0.0
      IF(I.LE.K1)GO TO 50
      GO TO 520
C*************************
C***  COSTS HERE ARE IN FORM OF ONE COST BUT NEED TO BE INFLATED
C***  COST (1) IS INFLATED BY YEAR
C*************************
  250       IF (COST(1) .EQ. 0.0 .AND. ISA(1) .NE. 0) GO TO 270
           DO 255 J=1,NYR
      JJ=J+JYDSC
  255      FLCOST(J)=COST(1)*FLAT**JJ
        WRITE(8'ISUB(1),ERR=600) ISA,CODE,TITLE,COST,
     +FLCOST
           GO TO 300
  270      I1=2
  275   IF(ISA(I1) .LE. 0) GO TO 290
           READ(8'ISA(I1),ERR=600) ISB,CODE,TITLE,COST
      IF (COST(1) .EQ. 0) GO TO 300
        DO 280 J=1,NYR
      JJ=J+JYDSC
  280   FLCOST(J)=COST(1)*FLAT**JJ
      WRITE(8'ISA(I1),ERR=600) ISB,CODE,TITLE,COST,
     +FLCOST
  290   I1=I1+1
            IF(I1 .LE. K1) GO TO 275
  300    I=I+1
      CHK=0.0
      IF(I .LE. K1) GO TO 50
      GOTO 520
  350   DO 370 JJ=1,NYR
C*************************
C***  THESE COSTS ARE NOT INFLATED BUT COSTS ARE PLACED IN FLCOST
C*************************
      IF(ISWTCH .NE. 6)FLCOST(JJ)=COST(JJ)
      FLCOST(JJ)=COST(1)
  370   CONTINUE
      WRITE(8'ISUB(I),ERR=600) ISA,CODE,TITLE,COST,
     +FLCOST
```

```
   112              GO TO 300
   113        520   RETURN
   114        600   WRITE(6,601)
   115        601   FORMAT(5X,'ERROR IN N=LATE')
   116              RETURN
   117        C      DEBUG TRACE,SUBTRACE,INIT
   118        C      AT 1
   119        C      TRACE ON
   120              END
```

```
@PRT,S A.PASES
```

SH*COST(1).PAGES

```
      1              SUBROUTINE PAGES(NUML)
      2              COMMON/PAGE/N1,IPAGNO,LCN,LNCNT,TITLE
      3              COMMON/SEC/ISECT
      4              COMMON/CITYS/NAMES(151)
      5              CHARACTER NAMES*63,N1*6,TITLE*30,NAME*63
      6              CHARACTER*27 FORMTS(9),FORMAT
      7              CHARACTER*4 MONTHS(12)
      8              DATA NAME/63H
      9             1                                  /
     10              DATA MONTHS/'JAN.','FEB.','MAR.','APR.','MAY ','JUNE','JULY',
     11             1            'AUG.','SEP.','OCT.','NOV.','DEC'/
     12              DATA FORMTS/'(1H+,T120,I3,1H.,I1,1H.,I1)',
     13             1            '(1H+,T120,I3,1H.,I1,1H.,I2)',
     14             1            '(1H+,T120,I3,1H.,I1,1H.,I3)',
     15             1            '(1H+,T120,I3,1H.,I2,1H.,I1)',
     16             1            '(1H+,T120,I3,1H.,I2,1H.,I2)',
     17             1            '(1H+,T120,I3,1H.,I2,1H.,I3)',
     18             1            '(1H+,T120,I3,1H.,I3,1H.,I1)',
     19             1            '(1H+,T120,I3,1H.,I3,1H.,I2)',
     20             1            '(1H+,T120,I3,1H.,I3,1H.,I3)'/
     21              LNCNT=LNCNT+NUML
     22              IF(LNCNT.LE.53) RETURN
     23              IPAGNO=IPAGNO+1
     24              LNCNT=NUML
     25              IF(LNCNT.GT.55) LNCNT=9
     26              DECODE(6,101,N1) IM,ID,IY
     27      101     FORMAT(3I2)
     28                  IF(LCN.NE.0) NAME=NAMES(LCN)
     29              WRITE(6,100) TITLE,NAME,MONTHS(IM),ID,IY
     30              NUM=0
     31              IF(ISECT.GE.10) NUM=NUM+3
     32              IF(ISECT.GE.100) NUM=NUM+3
     33              IF(IPAGNO.GE.10) NUM=NUM+1
     34              IF(IPAGNO.GE.100) NUM=NUM+1
     35              NUM=NUM+1
     36              FORMAT=FORMTS(NUM)
     37              WRITE(6,FORMAT) LCN,ISECT,IPAGNO
     38      100     FORMAT(1H1,A30,A63,1X,A4,I3,' P',I2)
     39              WRITE(6,102)
     40      102     FORMAT(//)
     41              RETURN
     42              END
```

aPRT,S A.PARS

SH*COST(1).PARS

```
      1       PARS  PROC
      2       C  PARAMETERS FOR SOLAR COST PROGRAM
      3       C           PARAMETERS IS1 & IC1 MUST BE UPDATED IF THE BLOCK # OF
      4       C          SHVAC AND/OR CHVAC ARE CHANGED. OTHER BLOCKS UNDER
      5       C          THESE BLOCKS NEED NOT BE CHANGED AS LONG AS THEY CORRESPOND
      6       C           TO THE BLOCK # INDICATED IN THIS PROC. FAILURE TO ADJUST THESE
      7       C           PARAMETERS TO THE APPROPRIATE VALUES WILL YIELD UNPREDICTABLE
      8       C          RESULTS. THESE PARAMETERS ARE USED TO KEY ON THESE BLOCKS
      9       C          FOR THE SPECIFIC CALCULATIONS USED FOR THAT ITEM.
     10       C
     11                  PARAMETER ILFE=100     @BLOCK # OF LIFE CYCLE COSTS
     12       C           SOLAR PARAMETERS ISX
     13             PARAMETER IS1=1    @ BLOCK # OF SHAC
     14             PARAMETER IS2=IS1+1        @ BLOCK OF ACQUISITIONS
     15             PARAMETER IS3=IS1+2   @ BLOCK # OF BUILDING MODIFICATIONS
     16             PARAMETER IS4=IS1+3      @ BLOCK # OF SPACE OCCUPIED
     17             PARAMETER IS5=IS1+4      @ BLOCK # OF NET REPAIRS
     18             PARAMETER IS6=IS1+5      @ BLOCK # OF MAINTENANCE
     19             PARAMETER IS7=IS1+6      @ BLOCK # OF OPERATIONS
     20             PARAMETER IS8=IS1+7      @ BLOCK # OF PROPERTY TAX
     21             PARAMETER IS9=IS1+8      @ BLOCK # OF PROPERTY TAX DEDUCTIONS
     22             PARAMETER IS10=IS1+9     @ BLOCK # OF DEPRECIATIN DEDUCTIONS
     23             PARAMETER IS11=IS1+10    @ BLOCK # OF ADDED INCOME
     24             PARAMETER IS12=IS1+11    @ BLOCK # OF INSURANCE COSTS
     25             PARAMETER IS13=IS1+12    @ BLOCK # OF SALVAGE VALUE AFTER N YEARS
     26             PARAMETER IS14=IS1+13    @ BLOCK # OF LOAN MORTAGE
     27             PARAMETER IS15=IS1+14         @ BLOCK # OF LOAN PAY
     28              PARAMETER IS16=IS1+15        @ BLOCK # OF MAINTENANCE DEDUCTS FROM INC TAX
     29                 PARAMETER IS17=IS1+16         @ BLOCK # OF ENERGY DEDUCTS FOR COMMERCIAL
     30       C  CONVENTIONAL PARAMETERS ICX
     31             PARAMETER IC1=101          @ BLOCK # OF CHVAC
     32             PARAMETER IC2=IC1+1        @ BLOCK # OF ACQUISITIONS
     33             PARAMETER IC3=IC1+2        @ BLOCK # OF BUILDING MODIFICATIONS
     34             PARAMETER IC4=IC1+3        @ BLOCK # OF SPACE OCCUPIED
     35             PARAMETER IC5=IC1+4        @ BLOCK # OF NET REPAIRS
     36             PARAMETER IC6=IC1+5        @ BLOCK # OF MAINTENANCE
     37             PARAMETER IC7=IC1+6        @ BLOCK # OF OPERATIONS
     38             PARAMETER IC8=IC1+7        @ BLOCK # OF PROPERTY TAX
     39             PARAMETER IC9=IC1+8        @ BLOCK # OF PROPERTY TAX DEDUCTIONS
     40             PARAMETER IC10=IC1+9       @ BLOCK # OF DEPRECIATION DEDUCTIONS
     41             PARAMETER IC12=IC1+12      @ BLOCK # OF SALVAGE
     42             PARAMETER IC11=IC1+11      @ BLOCK # OF INSURANCE
     43             PARAMETER IC13=IC1+13      @ BLOCK # OF MORTAGE
     44             PARAMETER IC14=IC1+14         @ BLOCK # OF LOAN PAY
     45              PARAMETER IC15=IC1+15
     46                 PARAMETER IC16=IC1+16
     47             PARAMETER IY=20       @ ARRAY SIZE OF YEAR ARRAY
     48             PARAMETER NCST=IY+8       @ARRAY SIZE OF COSTS ARRAYS
     49             PARAMETER  NTOT=NCST   @ ARRAYSIZE OF TOTAL
     50             PARAMETER K1=17            @ARRAY SIZE IF ISUB
     51             PARAMETER IP1=2           @ ARRAYSIZE OF CODE
     52             PARAMETER IP2=6           @ ARRAY SIZE OF TITLE
     53             PARAMETER K2=2*NCST+2*NTOT+K1+IP1+IP2+8
     54             PARAMETER  K3=K2+2+25        @ BUFFER ARRAY SIZE
     55             PARAMETER L1=2           @ # OF LEVEL 1 GROUPS
```

```
      PARAMETER L2=8          @ # OF LEVEL 2 GROUPS
      PARAMETER L3=3            @ # OF LEVEL 3 GROUPS
      PARAMETER L4=0          @ # OF LEVEL 4 GROUPS
      PARAMETER   L5=0         @ # OF LEVEL 5 GROUPS
      PARAMETER  L6=10          @ # OF LEVEL6 GROUPS
      PARAMETER  NR=L1+L2+L3+L4+L5+L6
      PARAMETER IGFL=25
      PARAMETER NTR=NR+4
          INTEGER DATOPT,CLTPRM
      CHARACTER*5 CODE,CCODE,ECODE,ACODE,BCODE,DCODE
      CHARACTER*4 TITLE(6),ATITLE(6),BTITLE(6),CTITLE(6),DTITLE(6)
     +,ETITLE(6)
      COMMON/SHAC2/ISUB(K1),CODE,TITLE,COST(MCST),TOTAL(MTOT),
     +FLCOST(MCST),AFCOST(MCST),BFCOST(MCST),CFCOST(MCST),DFCOST(MCST),
     +ACTUAL(MTOT),BCTUAL(MTOT),CCTUAL(MTOT),DCTUAL(MTOT)
      COMMON/SHAC1/NYR,IYR(IY),DISC,ACFACT,ICOM,IRC1,IRC2,IRC3,IRC4,
     +NBLK,BUFFER(K3),IYRST,I,IJK,IRC5,IDSC,INRGY,ILFCY,JYDSC,IDISC,
     +DUMPRT,CSTPSM,CLTPRM,LCN,LOA,XLOAD,IPLOTS,TINCOM,PTAX
      COMMON/SHAC3/  ACOST(MCST),BCOST(MCST),CCOST(MCST),DCOST(MCST),
     +ATITLE,BTITLE,CTITLE,DTITLE,ACODE,DATOPT,BCODE,
     +CCODE,DCODE,ISA(K1),ISB(K1),ISC(K1),ISD(K1),ATOTAL(MTOT),
     +BTOTAL(MTOT),CTOTAL(MTOT)
      COMMON/GFLD/AREA(IGFL),PCT(IGFL),G(IGFL),PRICE(IGFL)
      COMMON/DSFTOP/DSSTRT,DSSTOP,DSSTEP,FTSTRT,FTSTOP,FTSTEP,
     +DPSTRT,DPSTOP,DPSTEP,IPLTOP,PTSTRT,PTSTOP,PTSTEP,TSTOP,TSTART
     +,TSTEP
          DIMENSION ISE(K1),ETOTAL(MCST),ECOST(MCST)
      END

$PRT,S A.PLTDRV
```

SH*COST(1).PLTDRV

```
      1            SUBROUTINE PLTDRV(X,Y,XCNT,IPLTOP,IPLOTS)
      2            INTEGER XCNT
      3            CHARACTER*55 LABEL(6),LAB(5)
      4            DIMENSION X(25),Y(25,5),Z(25),K(5)
      5
      6      C     PRINT *,'ONE'
      7      50    K(I)=0
      8            DO 10 J=2,5
      9            YMAX=Y(1,J)
     10            DO 13 I=2,XCNT
     11      10    IF(Y(I,J).GT.YMAX) YMAX=Y(I,J)
     12      11    IF(YMAX.LT.100.0) GO TO 12
     13            K(J)=K(J)+1
     14            YMAX=YMAX/10.0
     15            GO TO 11
     16      12    DO 14 I=1,XCNT
     17      14    Y(I,J)=Y(I,J)/(10**K(J))
     18            LABEL(4)=50HPRESENT VALUE COST
     19            LABEL(2)=50HACTUAL COST
     20            IMIN=X(1)
     21            XMAX=FLOAT(IMIN)
     22            XMIN=XMAX
     23            IF(X(XCNT)-XMAX+0.) 19,16,17
     24      19    XMAX=XMAX+.8
     25            IF(XMAX.LT.X(XCNT)) GO TO 19
     26            GO TO 18
     27      16    XMAX=X(XCNT)
     28            GO TO 18
     29      17    XMAX=XMAX+8.
     30            IF(XMAX.LT.X(XCNT)) GO TO 17
     31      18    YMIN=0.
     32            YMAX=9).
     33            GO TO (1,2,3,4,5,6),IPLTOP
     34      1     LABEL(1)=50HCOLLECTOR AREA
     35            LABEL(2)=50HPERCENT LOAD - ACTUAL COST
     36            LABEL(3)=50HPERFORMANCE AND COST AS RELATED TO COLLECTOR AREA
     37            LABEL(5)=50HPRESENT VALUE COST AS RELATED TO COLLECTOR AREA
     38      C      XMAX=160.0
     39      C      XMIN=0.
     40            YMIN=1.
     41            YMAX=100.
     42            GO TO 9
     43      2     LABEL(1)=50HDISCOUNT RATE
     44            LABEL(3)=50HACTUAL COST AR RELATED TO DISCOUNT RATE
     45            LABEL(5)=50HPRESENT VALUE COST AS RELATED TO DISCOUNT RATE
     46            GO TO 9
     47      3     LABEL(1)=50HINFLATION RATE
     48            LABEL(3)=50HACTUAL COST AS RELATED TO INFLATION RATE
     49            LABEL(5)=50HPRESENT VALUE COST AS RELATED TO INFLATOON RATE
     50            GO TO 9
     51      4     LABEL(1)=50HDOWNPAYMENT PERCENT
     52            LABEL(5)=50HPRESENT VALUE COST AS RELATED TO DOWNPAYMENT
     53            LABEL(3)=50HACTUAL COST AS RELATED TO DOWNPAYMENT
     54            GO TO 9
     55      5     LABEL(1)=50HPROPERTY TAX RATE
```

```
      LABEL(3)=50HACTUAL COST AS RELATED TO PROPERTY TAX RATE
      LABEL(5)=50HPRESENT VALUE COST AS RELATED TO PROPERTY TAX RATE
      GO TO 9
    6 LABEL(1)=50HINCOME TAX RATE
      LABEL(3)=50HACTUAL COST AS RELATED TO INCOME TAX RATE
      LABEL(5)=50HPRESENT VALUE COST AS RELATED TO INCOME TAX RATE
    9 CONTINUE
      IF(IPLOTS.EQ.2) GO TO 30
      DO 21 J=1,3
      DO 20 I=1,XCNT
   20 Z(I)=Y(I,J)
      ICODE=J
      IF(IPLTOP.NE.1.AND.J.EQ.1) GO TO 21
      IF(IPLTOP.NE.1.AND.J.EQ.2) ICODE=1
      CALL PLOTER (X,Z,XCNT,LABEL(1),LABEL(2),LABEL(3),XMIN,XMAX,YMIN,
     $YMAX,ICODE)
   21 CONTINUE
      LAB(1)=50HPRECENT LOAD COVERED
      LAB(2)=50HACTUAL COST OF THE SHVAC
      LAB(3)=50HACTUAL COST OF THE CHVAC
      LAB(4)=50HPRESENT VALUE COST OF THE SHVAC
      LAB(5)=50HPRESENT VALUE COST OF THE CHVAC
      ICODE=1
      IF(IPLTOP.NE.1) GO TO 51
      CALL LEGEND(LAB(1),K(1),1)
      ICODE=2
   51 CALL LEGEND(LAB(2),K(2),ICODE)
      CALL LEGEND(LAB(3),K(3),3)
   30 DO 53 J=4,5
      DO 54 I=1,XCNT
   54 Z(I)=Y(I,J)
      ICODE=1
      IF(J.EQ.5) ICODE=3
   53 CALL PLOTER(X,Z,XCNT,LABEL(1),LABEL(4),LABEL(5),XMIN,XMAX,YMIN,
     $YMAX,ICODE)
      CALL LEGEND(LAB(4),K(4),1)
      CALL LEGEND(LAB(5),K(5),3)
       DO 15 I=1,XCNT
      DO 15 J=1,5
   15 Y(I,J)=0
      XCNT=0
      RETURN
      END
```

$PRT,S A.PLOTS

SH*COST(1).PLOTS

```
      1            SUBROUTINE PLOTER(A,B,NR,XTLE,YTLE,HEADER,XMIN,XMAX,YMIN,YMAX,ID)
      2            DIMENSION X(25),Y(25),A(25),B(25)
      3            CHARACTER*4 XTLE(16),YTLE(16),HEADER(16),LOCN(16)
      4            IHCHR=0
      5            IXCHR=0
      6            IYCHR=0
      7            DO 15 I=1,16
      8            IF(XTLE(I).EQ.4H      ) GO TO 11
      9       10   IXCHR=IXCHR+4
     10       11   DO 12 I=1,16
     11            IF(YTLE(I).EQ.4H      ) GO TO 13
     12       12   IYCHR=IYCHR+4
     13       13   DO 14 I=1,16
     14            IF(HEADER(I).EQ.4H      ) GO TO 15
     15       14   IHCHR=IHCHR+4
     16       15   CONTINUE
     17            IFLG=0
     18            DO 16 II=1,NR
     19            X(II)=A(II)
     20       16   Y(II)=B(II)
     21            IF(ID.NE.1) GO TO 25
     22            XNUM=0
     23            CALL SYMBOL(0.0,10.2,.143,HEADER,0.0,IHCHR)
     24            SCA=(XMAX-XMIN)/8.0
     25            CALL AXIS(0.0,1.0,XTLE,-IXCHR,8.0,0.0,XMIN,SCA)
     26            SCA=(YMAX-YMIN)/9.0
     27            CALL AXIS(0.0,1.0,YTLE,IYCHR,90.0,90.0,YMIN,SCA)
     28            DO 23 I=0,8
     29            XB=FLOAT(I)
     30            CALL PLOT(XB,1.0,3)
     31            CALL PLOT(XB,10.0,2)
     32            IF(I.EQ.8) GO TO 23
     33            CALL PLOT(XB+0.5,10.0,3)
     34            CALL PLOT(XB+0.5,1.0,2)
     35       23   CONTINUE
     36            DO 24 I=1,10
     37            XB=FLOAT(I)
     38            CALL PLOT(0.0,XB,3)
     39            CALL PLOT(8.0,XB,2)
     40            IF(I.EQ.10) GO TO 24
     41            CALL PLOT(8.0,XB+0.5,3)
     42            CALL PLOT(0.0,XB+0.5,2)
     43       24   CONTINUE
     44       25   CALL SCALE(X,XMIN,XMAX,0.0,8.0,NR)
     45            CALL SCALE(Y,YMIN,YMAX,1.0,9.0,NR)
     46       31   CALL PLOT(X(1),Y(1),3)
     47            DO 33 K1=2,NR
     48            IF(X(K1).LT.0.0) X(K1)=0.0
     49            IF(X(K1).GT.9.0) X(K1)=9.0
     50            IF(Y(K1).LT.0.0) Y(K1)=0.0
     51            IF(Y(K1).GT.10.0) Y(K1)=10.0
     52            CALL PLOT(X(K1),Y(K1),2)
     53       33   CONTINUE
     54            XNUM=XNUM+1
     55            DO 32 K1=1,NR
```

```
     56       32    CALL NUMBER(X(K1),Y(K1),.37[,XNUM,),0,-3)
     57             IF(ID.EQ.3) CALL PLOT(12.0,0.0,-3)
     58             RETURN
     59             END
```

```
@PRT,S A.POSAVL
```

SH•COST(1).POSAVE

```
      1              SUBROUTINE POSAVE
      2              INCLUDE PARS
      3              COMMON/SEC/ISECT
      4              COMMON/IPL/IP
      5              COMMON/NFLT/FLATE
      6              COMMON/CYLIF/CYLF(6)
      7              COMMON/SUM/SHVAC,CHVAC,ACLCS,PVLCS,ACLCFC,PVLCFC,PSHAC,PCHAC
      8              DIMENSION CUMSAV(IY),ACUMSV(IY)
      9              READ(8'IS1,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,TOTAL,
     10             •ACTUAL
     11              READ(8'IC1,ERR=600) ISE,ECODE,ETITLE,ECOST,ETOTAL,ATOTAL,
     12             •ACTUAL
     13                   IFLG=1
     14                   IAFLG=1
     15              DO 100 I=1,NYR
     16               II=I-1
     17              APOSAV=BCTUAL(I)-ACTUAL(I)
     18              IF(I.NE.1)ACUMSV(I)=ACUMSV(II)+APOSAV
     19              IF(I.EQ.1)ACUMSV(I)=APOSAV
     20              PSAVE=ATOTAL(I)-TOTAL(I)
     21              IF(I.NE.1)CUMSAV(I)=CUMSAV(II)+PSAVE
     22              IF(I.EQ.1)CUMSAV(I)=PSAVE
     23              IF(APOSAV.LE. 0.OR.IAFLG .NE. 1) GO TO 70
     24                   IAFLG=99
     25                   IASVYR=I+IYRST
     26        70    IF(PSAVE .LE. 0.OR. IFLG .NE. 1) GO TO 100
     27                   IFLG=99
     28                   ISVYR=I+IYRST
     29        100   CONTINUE
     30              ISV=ISVYR-IYRST
     31              IASV=IASVYR-IYRST
     32              FSAVE=ATOTAL(ISV)-TOTAL(ISV)
     33              APOSAV=BCTUAL(IASV)-ACTUAL(IASV)
     34              CALL PAGES(NYR+8)
     35                   WRITE(6,200) ISVYR,PSAVE,IASVYR,APOSAV,(IYR(K),CUMSAV(K),
     36             •ACUMSV(K),K=1,NYR)
     37              FLATE1=FLATE•100.0
     38              TINCR1=TINCOR•100.0
     39              PTAX1=PTAX•100.0
     40              DISC1=DISC•100.0
     41              DNPMT=DNPMT•100.0
     42              IF(IPLTOP.EQ.1) VARY=AREA(IP)
     43              IF(IPLTOP.EQ.2) VARY=DISC•100.0
     44              IF(IPLTOP.EQ.3) VARY=FLATE•100.0
     45              IF(IPLTOP.EQ.4) VARY=DWNPMT•100.0
     46              IF(IPLTOP.EQ.5) VARY=PTAX•100.)
     47              IF(IPLTOP.EQ.6) VARY=TINCOR•100.0
     48              WRITE(9'ISECT,ERR=600) ISECT,VARY,PCT(IP),3(IP)
     49             •,FLATE1,DISC1,DNPMT,CSTPSM,TINCR1,PTAX1,AREA(IP),SHVAC,CHVAC,CYLF
     50             •,PSHAC,PCHAC
     51        200   FORMAT(1X,132('•'),/,2X,'IN YEAR ',I5,1X,'POSITIVE SAVINGS WILL ',
     52             •'OCCUR BY',/,2X,'AN AMOUNT OF ',F10.2,' IN TERMS OF PRESENT VALUE',
     53             •'DOLLARS',/,2X,'IN YEAR ',I5,1X,'POSITIVE SAVINGS WILL OCCUR',
     54             •' BY',/,2X,'AN AMOUNT OF ',F10.2,' IN TERMS OF ACTUAL DOLLARS',//,
     55             •2X,'CUMULATIVE SAVINGS BY YEARS',/,2X,'YEARS',2X,'PRESENT
```

```
     56              +' VALUE SAVINGS',2X,'ACTUAL SAVINGS',/,
     57              +80('-'),/,50(1X,I5,8X,F10.2,12X,F10.2,/))
     58               RETURN
     59        600        WRITE(6,601)
     60        601        FORMAT(5X,'ERROR IN POSAVE')
     61                   RETURN
     62               END

@PRT,S A.PVAC
```

```
SH*COST(1).PVAC
      1              SUBROUTINE PVAC
      2              INCLUDE PARS
      3              DIMENSION IBTRM(13)
      4              DATA IBTRM/9,10,13,14,11,16,17,109,110,112,113,115,116/
      5        C   COMBINES TOTALS IN MAJOR AREAS TO OBTAIN PV AND AC
      6              IF(IRC1.EQ.0) GO TO 150
      7              CALL SBLNK
      8              READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
      9             +FLCOST
     10              KK=K1+10+MCST*2
     11                  AC=0.0
     12                  PV=0.0
     13              ACT=0.0
     14              NYRR=NYR+4
     15              DO 100 I=2,K1
     16              SIGN=1.0
     17               IF(ISUB(I).LE.0) GOTO 130
     18        C     READ SUBLEVELS FOR PREVIOUSLY COMPUTED PV AND AC COSTS
     19              READ(8'ISUB(I),ERR=630) ISA,BCODE,BTITLE,BCOST,
     20             +BFCOST,TOTAL,ACTUAL
     21        C     DETERMINE SIGN FOR CALCULATION
     22        C*** PROPERTY TAX DEDUCTIONS, DEPRECIATION DEDUCTIONS, SALVAGE
     23        C*** VALUE AFTER N YEARS,  INTEREST ON MORTGAGE AND
     24        C*** AFTER TAX INCOME ARE SUBTRACTED FROM THE TOTAL COSTS
     25        C*** ALL OTHER COSTS ARE ADDED TO THE TOTAL COSTS.
     26              DO 10 II = 1,13
     27        10    IF(ISUB(I).EQ.IBTRM(II)) SIGN=-1.0
     28              IF(ISUB(I) .GT. IRC1+3 )GO TO 35
     29              ATOTAL(1)=TOTAL(NYR+3)*SIGN+ATOTAL(1)
     30              BCTUAL(1)=TOTAL(NYR+3)*SIGN+BCTUAL(1)
     31              ACT=ACT+TOTAL(NYR+3)*SIGN
     32              PV=PV+TOTAL(NYR+3)*SIGN
     33              AC=AC+TOTAL(NYR+2)*SIGN
     34              GO TO 100
     35        35    IF(ISUB(I).NE.13.AND.ISUB(I).NE.112) GO TO 37
     36              ACT=ACT-ACTUAL(NYR+1)
     37              PV=PV-TOTAL(NYR+1)
     38              AC=AC-TOTAL(NYR+2)
     39              BCTUAL(NYR)=BCTUAL(NYR)-ACTUAL(NYR+1)
     40              ATOTAL(NYR)=ATOTAL(NYR)-TOTAL(NYR+1)
     41              GO TO 130
     42        37    DO 40 J=1,NYR
     43              ATOTAL(J)=TOTAL(J)*SIGN+ATOTAL(J)
     44              BCTUAL(J)=ACTUAL(J)*SIGN+BCTUAL(J)
     45        40    CONTINUE
     46        70    ACT=ACT+ACTUAL(NYR+1)*SIGN
     47              PV=PV+TOTAL(NYR+1)*SIGN
     48              AC=AC+TOTAL(NYR+2)*SIGN
     49        100   CONTINUE
     50              IF(INRGY.NE.1) GO TO 140
     51              ACT=COST(3)+ACT
     52              AC=AC+COST(2)
     53              PV=PV+COST(1)
     54        C   WRITE TOTAL PV AND AC INTO TOP LEVEL
     55        140   WRITE(8'IRC1,ERR=620) ISUB,CODE,TITLE,COST,
```

```
     56              *FLCOST,ATOTAL,BCTUAL,PV,AC,ACT
     57        150   RETURN
     58        600   WRITE(6,601)
     59        601   FORMAT(5X,'ERROR IN PVAC')
     60              RETURN
     61              END

APRT,S A.RESULT
```

SH*COST(1).RESULT

```
      1             SUBROUTINE RESULT
      2             INTEGER XCNT
      3             COMMON/XC/XCNT
      4             COMMON/FLAG/IFLG
      5             COMMON/NFLT/FLATE
      6             COMMON/IP1/IP
      7             COMMON/SUM/SHVAC,CHVAC,ACLCS,PVLCS,ACLCFC,PVLCFC,PSHAC,PCHAC
      8             COMMON/SEC/ISECT
      9             COMMON/LAST/PRICEL
     10             INCLUDE PARS
     11             COMMON/CYLIF/CYLF(6)
     12             COMMON/LNCM/NUM
     13             DIMENSION X(25),Y(25,5)
     14             DIMENSION NOUT(17)
     15             DATA NOUT/3,3,4,4,9,9,9,9,9,9,6,6,6,9,9,9,9/
     16             IRC1=ISI
     17             IF(IRC1.EQ.0) GO TO 250
     18        C   PRINTS OUT RESULTS
     19        100    CALL SBLNK
     20               READ(8'IRC1,ERR=600) ISUB,CODE,TITLE
     21               CALL PAGES(99)
     22                   CALL PAGES(3)
     23               WRITE(6,10) CODE,TITLE
     24               DO 200 I=2,K1
     25        C*** READ SUBLEVELS OF TOP LEVEL
     26                 IF(ISUB(I).LE.0) GO TO 200
     27               KL=ISUB(I)
     28               IF(KL.GT.110) KL=KL+1
     29               IF(KL.GT.100) KL=KL-100
     30               NUM=NOUT(KL)+6
     31               IF(KL.GE.5.AND.KL.LE.10.OR.KL.GT.13) NUM=NUM+NYR
     32               CALL PAGES(NUM)
     33                 READ(8'ISUB(I),ERR=600) ISA,CODE,TITLE,ECOST,ETOTAL,
     34              +TOTAL,ACTUAL
     35        C      WRITE INTERMEDIATE RESULTS
     36                 CALL REPRNT
     37          200 CONTINUE
     38               READ(8'IRC1,ERR=600) ISE,CODE,TITLE,ECOST,ETOTAL,
     39              +TOTAL,ACTUAL,PV,AC,ACT
     40        C      WRITE FINAL RESULTS
     41                   CALL PAGES(NYR+8)
     42               WRITE(6,15) CODE,TITLE,PV,AC,ACT
     43               WRITE(6,18)(IYR(J),TOTAL(J),ACTUAL(J),J=1,NYR)
     44               CHVAC=ACT
     45               PCHAC=PV
     46        250    IF(IRC1.EQ.IC1)GO TO 300
     47        C*** CHANGE FROM SOLAR TO CONVENTIONAL
     48               IRC1=IC1
     49               SHVAC=ACT
     50               PSHAC=PV
     51               GO TO 100
     52        C***   WRITE LIFE CYCLE COSTS SAVINGS IF REQUESTED
     53        300    IF(ILFCY .NE. 1) GO TO 400
     54               READ(8'ILF,ERR=600) ISA,CODE,TITLE,CYLF
     55                   CALL PAGES(3)
```

```
      WRITE(6,305)
      CALL PAGES(9)
      WRITE(6,310)CYLF
      ACLCS=CYLF(2)
      PVLCS=CYLF(1)
      ACLCFC=CYLF(5)
      PVLCFC=CYLF(4)
305   FORMAT(1X,132('*'),/,1X,4X,'LIFE CYCLE COST SAVINGS')
310    FORMAT(1X,'PRESENT                    '
     +'      =',F10.2,/,1X,'ACTUAL                      ',7X,' =',
     +F10.2,/,1X,'ANNUALIZED                    =',F10.2,/,
     +1X,'THE PRESENT VALUE FUEL SAVINGS    =',F10.2,/,
     +1X,'THE ACTUAL FUEL SAVINGS           =',F10.2,/,1X,
     +'THE ANNUALIZED FUEL SAVINGS       =',F10.2,///)
 10   FORMAT(1X,132('*'),/,1X,A5,2X,'RESULTS FOR ',6A4,/)
 15  FORMAT(1X,2X,A5,1X,6A4,/,2X,'THE PRESENT VALUE COST OF THE SYSTEM'
     +' IS--',F12.2,/,1X,' THE ANNUALIZED COST OF THE SYSTEM IS--',
     +'---',F12.2,/,1X,' THE ACTUAL COST OF THE SYSTEM IS --------',
     +F12.2)
18    FORMAT(1X,2X,'COSTS BY YEAR',//,2X,'YEAR',5X,'PVCOST',5X,'ACTUAL',
     +/,2X,'----',5X,'------',5X,'------',//,
     +5(2X,I4,1X,F13.2,1X,F10.2,/))
400    XCNT=XCNT+1
       Y(XCNT,1)=PCT(IP)
       Y(XCNT,2)=SHVAC
       Y(XCNT,3)=CHVAC
       Y(XCNT,4)=PSHAC
       Y(XCNT,5)=PCHAC
       IF(IPLTOP.EQ.1) X(XCNT)=AREA(IP)
       IF(IPLTOP.EQ.2) X(XCNT)=DISC*100.0
       IF(IPLTOP.EQ.3) X(XCNT)=FLATE*100.0
       IF(IPLTOP.EQ.4) X(XCNT)=DWMPRT*100.0
       IF(IPLTOP.EQ.5) X(XCNT)=PTAX*100.0
       IF(IPLTOP.EQ.6) X(XCNT)=TINCOM*100.0
C          WRITE(6,1000) IFLG,IPLOTS
1000       FORMAT(5X,'IFLG',I5,' IPLOTS',I5)
       IF(IFLG.EQ.1.AND.IPLOTS.NE.0) CALL PLTDRV(X,Y,XCNT,IPLTOP,IPLOTS)
        RETURN
600    WRITE(6,601)
601    FORMAT(5X,'ERROR IN RESULT')
       RETURN
       END

$PRT,S A.RFPRNT
```

```
SH*COST(1).RFPRNT
      1            SUBROUTINE RFPRNT
      2            INCLUDE PARS
      3            COMMON/LNCM/NUM
      4            IF(ISUB(I).EQ.0) GO TO 200
      5      C     SUBLEVEL HEADING
      6            CONTINUE
      7            WRITE(6,5) CODE,TITLE,NYR
      8          5 FORMAT(1X,///,40X,A5,2X,6A4,/,2X,'COSTS OVER THE ',I3,' YEARS OF'
      9           +,' THE ANALYSIS')
     10            ISWTCH=ISUB(I)-1
     11            IF(ISWTCH .GT. 100) GO TO 1
     12            GO TO(1000,10,10,50,50,50,40,40,40,10,10,10,40,40,40,40,40),ISWTCH
     13      1     GOTO(1000,10,10,50,50,50,40,40,40,10,10,40,40,40,40,40),ISWTCH-100
     14      C     WRITE COST RESULTS
     15      10    CONTINUE
     16            WRITE(6,16) TOTAL(NYR+1),TOTAL(NYR+2),ACTUAL(NYR+1)
     17      17    FORMAT(1X,//,1X,'THE DOWN PAYMENT =',1X,F10.2,/,1X,F5.2,'% OF ',
     18           +'THE TOTAL COST')
     19            IF(ISA(2) .GT. 0) GO TO 12
     20            GO TO 200
     21      12    CALL PAGES(NUM+4)
     22            WRITE(6,9)
     23      9     FORMAT(1X,2X,'SUBLEVEL COSTS',///)
     24            DO 14 JJ=2,K1
     25            IF(ISA(JJ).LE. 0) GO TO 14
     26            READ(8'ISA(JJ),ERR=600) ISE,ACODE,ATITLE,ECOST,ETOTAL,
     27           +BCTUAL,ATOTAL
     28            WRITE(6,15) ACODE,ATITLE,ATOTAL(NYR+1),BCTUAL(NYR+1)
     29      14    CONTINUE
     30         15 FORMAT(1X,A5,6A4,/,1X,2X,'PRESENT VALUE = ',F12.2,/,
     31           +3X,'ACTUAL COST    =',F12.2,/)
     32      1000  DOWNPR=DWNPMT*100.0
     33            WRITE(6,17)TOTAL(NYR+3),DOWNPR
     34            GO TO 200
     35         16 FORMAT(1X,2X,'PRESENT VALUE = ',F12.2,/,3X,'ANNUAL ',
     36           +'COST    = ',F12.2,/,3X,'ACTUAL COST    =',F12.2,/)
     37      C     WRITE COST RESULTS WITH YEARLY PV RESULTS
     38      40    WRITE(6,45) TOTAL(NYR+1),TOTAL(NYR+2),ACTUAL(NYR+1),(IYR(J),
     39           +TOTAL(J),ACTUAL(J),J=1,NYR)
     40         45 FORMAT(1X,2X,'PRESENT VALUE = ',F12.2,/,3X,'ANNUAL COST   = ',
     41           +F12.2,/,3X,'ACTUAL COST    = ',F12.2,//,3X,'PRESENT VALUE AND ',
     42           +'ACTUAL COSTS BY YEAR',//,4X,
     43           +'YEAR',5X,'PVCOST',5X,'ACTUAL',/,4X,'----',5X,'------',5X,'------'
     44           +,/,50(3X,I5,1X,F10.2,1X,F10.2,/))
     45            GO TO 200
     46      C*** WRITE YEARLY COSTS FOR THIS ITEM AND YEARLY COSTS FOR EACH
     47      C*** OF THE COMPONENTS COMPRISING THIS ITEMS COSTS
     48      50    WRITE (6,45) TOTAL(NYR+1),TOTAL(NYR+2),ACTUAL(NYR+1),(IYR(K),
     49           +TOTAL(K),ACTUAL(K),K=1,NYR)
     50            IF(ISA(2) .LE. 0) GO TO 200
     51            CALL PAGES(NUM+6)
     52            WRITE(6,220) CODE,TITLE
     53            DO 100 J=2,K1,3
     54            J1=J+1
     55            J2=J+2
```

```
      IF(ISA(J).EQ.0)GO TO 100
      IF(J.NE.2) CALL PAGES(NUM)
        READ(8'ISA(J),ERR=600) ISE,CODE,TITLE,ECOST,ETOTAL,
     +TOTAL,ACTUAL
      IF(ISA(J1).LE.0)GO TO 70
        READ(8'ISA(J1),ERR=600) ISE,ACODE,ATITLE,ECOST,ETOTAL,
     +ATOTAL,BCTUAL
      IF(ISA(J2).LE.0) GO TO 60
        READ(8'ISA(J2),ERR=600) ISE,BCODE,BTITLE,ECOST,ETOTAL,
     +BTOTAL,CCTUAL
        WRITE (6,205) CODE,(TITLE(KA),KA=1,5),ACODE,(ATITLE(KB),KB=1,5),
     +BCODE,(BTITLE(KC),KC=1,5)
        WRITE (6,210) TOTAL(NYR+1),ATOTAL(NYR+1),BTOTAL(NYR+1),
     +TOTAL(NYR+2),ATOTAL(NYR+2),BTOTAL(NYR+2),ACTUAL(NYR+1),
     +BCTUAL(NYR+1),CCTUAL(NYR+1),(IYR(K),TOTAL(K),
     +ACTUAL(K),IYR(K),ATOTAL(K),BCTUAL(K),IYR(K),BTOTAL(K),
     +CCTUAL(K),K=1,NYR)
        GO TO 100
60    CONTINUE
      WRITE(6,206) CODE,(TITLE(KA),KA=1,5),ACODE,(ATITLE(KB),KB=1,5)
      WRITE(6,215) TOTAL(NYR+1),ATOTAL(NYR+1),TOTAL(NYR+2),
     +ATOTAL(NYR+2),ACTUAL(NYR+1),BCTUAL(NYR+1),(IYR(K),TOTAL(K),
     +ACTUAL(K),IYR(K),ATOTAL(K),BCTUAL(K),K=1,NYR)
        GO TO 100
70    CONTINUE
      WRITE(6,207)CODE,TITLE
        WRITE(6,85) TOTAL(NYR+1),TOTAL(NYR+2),ACTUAL(NYR+1),(IYR(K),
     +TOTAL(K),ACTUAL(K),K=1,NYR)
100   CONTINUE
      GO TO 200
150   WRITE(6,225) CODE,TITLE,ISUB(I),IRC1
  200 RETURN
600   WRITE(6,601)
601   FORMAT(5X,'ERROR IN REPRNT')
      RETURN
205   FORMAT(1X,/,3(2X,A5,1X,5A8,17X),/)
206   FORMAT(1X,/,2(2X,A5,1X,5A8,17X),/)
207   FORMAT(1X,/,2X,A5,1X,5A8,17X,/)
210   FORMAT(1X,/,3(3X,'PRESENT VALUE =',F12.2,19X),/,3(3X,'ANNUAL ',
     +'COST    =',F12.2,19X),/,3(3X,'ACTUAL COSTS  =',F12.2,19X),
     +///,3(3X,'COSTS BY YEAR',28X),
     +///,3(3X,'YEAR',5X,'PVCOST',5X,'ACTUAL',15X),/,3(3X,'----',5X,
     +'------',5X,'------',15X),///,50(3(2X,I5,1X,F10.2,1X,F10.2,15X),/),
     +/)
215   FORMAT(1X,/,2(3X,'PRESENT VALUE =',F12.2,19X),/,2(3X,'ANNUAL ',
     +'COST    =',F12.2,19X),/,2(3X,'ACTUAL COSTS  =',F12.2,19X),
     +///,2(3X,'COSTS BY YEAR ',27X),
     +///,3X,2(2X,'YEAR',5X,'PVCOST',5X,'ACTUAL',16X),/,3X,2(2X,'----',5X
     +,'------',5X,'------',16X),///,50(2(4X,I5,1X,F10.2,1X,F10.2,13X),/)
     +,/)
220   FORMAT(1X,3X,'PRESENT VALUE =',F12.2,/,3X,'ANNUAL COST    =',
     +F12.2,///,3X,'PRESENT VALUE COSTS BY YEAR',///,5X,
     +'YEAR',10X,'COST',/,5X,'----',10X,'----',///,
     +50(4X,I5,2X,F10.2,1X,F10.2,/))
222   FORMAT(1X,///,20X,'SUBLEVEL COSTS FOR ',A5,1X,6A8,///)
225   FORMAT(1X,/,3X,'**********ERROR IN ISUTCH IN REPRNT FOR',/,
```

```
    112              +3X,A5,1X,6A8,/,3X,'ISUB(I)=',I5,3X,'IRCI=',I5,//)
    113               END

@PRT,S A.SALVAG
```

```
SH*COST(1).SALVAG
      1              SUBROUTINE SALVAG
      2              INCLUDE PARS
      3        C    COMPUTE THE SALVAGE VALUE AFTER NYR
      4              IF(IRC1.EQ.0) GO TO 100
      5              READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST
      6              TOTAL(NYR+1)=COST(1)/(1.0+DISC)**NYR
      7              ACTUAL(NYR+1)=COST(1)
      8                TOTAL(NYR+2)=TOTAL(NYR+1)*ACTACT
      9              WRITE(8'IPC1,ERR=600) ISUB,CODE,TITLE,COST,
     10             *SLCOST,TOTAL,ACTUAL
     11        100    RETURN
     12        600    WRITE(5,601)
     13        601    FORMAT(5X,'ERROR IN SALVAG')
     14               RETURN
     15               END

$PRT,S A.SBLNK
```

SH*COST(1).SBLNK

```
      1              SUBROUTINE SBLNK
      2              INCLUDE PARS
      3        C   BLANKS OUT OR ZEROS
      4              CHARACTER*6 BLNK
      5              DATA BLNK/6H      /
      6              IF(IJK.EQ.0 ) GO TO 450
      7              DO 100 I=1,K1
      8         100    ISUB(I)=0
      9              DO 200 I=1,6
     10         200   TITLE(I)=BLNK
     11         300    CODE=BLNK
     12              DO 400 I=1,MCST
     13                COST(I)=0.0
     14                ACOST(I)=0.0
     15                BCOST(I)=0.0
     16                CCOST(I)=0.0
     17              FLCOST(I)=0.0
     18              AFCOST(I)=0.0
     19              BFCOST(I)=0.0
     20              CFCOST(I)=0.0
     21              DFCOST(I)=0.0
     22         400  DCOST(I)=0.0
     23         450 DO 500 I=1,MTOT
     24              ACTUAL(I)=0.0
     25               BCTUAL(I)=0.0
     26               CCTUAL(I)=0.0
     27                STOTAL(I)=0.0
     28                 TOTAL(I)=0.0
     29         500    ATOTAL(I)=0.0
     30              RETURN
     31              END
```

@PRT.S A.SKALE

SH•COST(1).SKALE

```
      1             SUBROUTINE SKALE(A,AMIN,AMAX,ORIG,ALNT,NUM)
      2             DIMENSION A(24)
      3             DO 10 I=1,NUM
      4        10   A(I)=(((A(I)-AMIN)/(AMAX-AMIN))•ALNT)+ORIG
      5             RETURN
      6             END
```

@PRT•S A.SOLMAN

SH*COST(1).SOLMAN

```
      INCLUDE PARS.LIST
      COMMON/SEC/ISECT
      COMMON/FLAG/IFLG
      COMMON /NFLT/ FLATE
      COMMON/PAGE/N1,IPAGNO,LCN1,LNCNT,NAME
      CHARACTER NAME*30
      CHARACTER*6 N1,N2,N3,N4
1     CONTINUE
      DEFINE FILE 8(200,K3,V,IDRR)
      DEFINE FILE 9(25,25,V,IDDRR1)
      CALL SOLOCK(N1,N2,N3,N4)
      READ(5,101) NAME
101   FORMAT(A30)
      IJK=1
C     CALL WBSIN FOR INPUT OF WBS STRUCTURE
      CALL WBSIN
C     CALL CSTDTN FOR INPUT OF COST DATA
      CALL CSTDTN
      IF(IPLOTS.NE.0) CALL CALLID
C*******************
C***  IDISC=1 INDICATES INFLATION BEGINS AFTER THE FIRST
C***  YEAR OF THE ANALYSIS IDISC=0 1 INDICATES FIRST YEAR
C***  INFLATION OF COSTS
C*******************
      JYDSC=0
      IF(IDISC .EQ.1) JYDSC=-1
C   COMPUTE SUCEEDING YEARS IN THE ANALYSIS
      DO 10 I=1,NYR
10    IYR(I)=IYRST+(I-1)
      LCN1=LCN
      ISECT=0
20    CALL DIVVY
      ISECT=ISECT+1
      LNCNT=0
      IF(DISC.EQ.-999.) GO TO 130
C************************
C***  IDSC=1 INDICATES DISCOUNT RATE IS REAL
C::   AND INFLATION RATE MUST BE ADDED TO DISC
C***  ELSE DISC IS REAL
C*******************
C************************
      IF(IDSC .EQ. 1) DISC=DISC+FLATE
C   COMPUTE THE FACTOR USED IN CONVERSION OF PRESENT VALUE TO ANNUAL COST
      ACFACT=(DISC*(1.0+DISC)**NYR)/((1.0+DISC)**NYR-1)
C     COMPUTE THE AMORTIZED VALUE OF THE LOAN FOR NYRS
      IF(INRGY .EQ. 1) GO TO 50
      IRC1=IS14
      IRC2=IS15
      CALL AMORTZ
      IRC1=113
      IRC2=114
      CALL AMORTZ
      IJK=0
C     COMPUTE THE INFLATED COSTS IF FLATE(INFLATION FACTOR) NOT=0
      IF (FLATE .EQ. 0.0) GO TO 50
```

```
      IPC1=IS1
      CALL NFLATE
      IRC1=IC1
      CALL NFLATE
C   CALL CSTDTO FOR OUTPUT OF COST DATA IN READABLE FORM
50    IF (DATOPT .NE. 1 ) CALL CSTDTO
C**************************
C***  IF ENERGY ANALYSIS ONLY FALL THRU
C***  ELSE CONTINUE NORMALLY BY CALLING SOLCST,CONCST
C**************************
      IF(INRGY .NE.1) GO TO 100
C**********************
C***  IF COMMERCIAL ENERGY DEDUCTS MUST BE COMPUTED
C***   ELSE GO TO ENERGY COSTS
C**********************
      IF(ICOM .NE.1) GO TO 90
      IRC1=0
      IPC2=0
      IRC3=0
      IPC4=IC16
      CALL DEPINC
      IRC4=IS17
      CALL DEPINC
C**************************
C***  ENERGY COSTS FOR ENERGY ONLY ANALYSIS
C**************************
90    IRC1=IC7
      CALL ENERGY
      IRC1=IS7
      CALL ENERGY
      IRC1=IS1
C**************************
C***  TOTAL RESULTS INTO PVAC
C**************************
      CALL PVAC
      IPC1=IC1
      CALL PVAC
      GOTO 120
C   CALL SOLCST AND CONCST FOR COMPUTATION OF COSTS FOR
C    BOTH THE SOLAR HVAC AND THE CONVENTIONAL HVAC
100   IF(IS1 .NE. 0)CALL SOLCST
      IF(IC1 .NE. 0)CALL CONCST
C**********************
C***  COMPUTE LIFE CYCLE COSTS IF REQUESTED
C**********************
120   IF(ILFCY .EQ. 1) CALL LIFCYC
C    CALL RESULT FOR OUTPUT OF THE RESULTS CALCULATED
      CALL RESULT
      CALL PDSAVE
      GO TO 20
130   CONTINUE
      CALL SUMRIZ(IPLTOP)
      STOP
C          DEBUG SUBTRACE,TRACE
C          AT 1
C          TRACE ON
```

```
    112             END

@PRT,S A.SHCOST
```

```
SH*COST(1).SHCOST
     1                  CHARACTER*80 DATA
     2                  CALL CSFREQ('@ASG,T 4. . ')
     3                  I=0
     4       10         READ(5,100,END=900) DATA
     5                    I=I+1
     6                  WRITE(6,200) I,DATA
     7       100        FORMAT(A80)
     8       200        FORMAT(5X,I5,5X,A80)
     9                  WRITE(4,100) DATA
    10                  GO TO 10
    11       900        CALL CSFREQ('@ADD SH*COST.BEGIN . ')
    12                  STOP
    13                  END

@PRT,S A.SOLCST
```

SH*COST(1).SOLCST

```
      1            SUBROUTINE SOLCST
      2            INCLUDE PARS.LIST
      3            DIMENSION IST(K1)
      4      C   CALLS SUBROUTINES FOR SUMMING LOWER LEVEL DATA, TOTALING DATA IN CERTAIN
      5      C   RECORDS, COMPUTING PRESENT VALUE AND ANNUAL COST FOR REP,MAINT,TAXES,ETC
      6            CONTINUE
      7      C    SUM LOWER LEVEL DATA TO HIGHER LEVEL
      8            READ(8'IS1,ERR=600) IST
      9            IF((IS1+IS2+IS3+IS4+IS5+IS6).EQ. 0) GO TO 30
     10            IRC1=IS1
     11            IRC2=IS6
     12      C     CALL SOLSUM
     13      C    TOTAL COSTS FOR ACQUISITIONS,BLDG MODS,AND SPACE OCCUPIED
     14      30    IF(IS2 .EQ. 0) GO TO 50
     15            IRC1=IS2
     16            CALL TOTALS
     17      50    IF(IS3 .EQ. 0) GO TO 70
     18            IRC1=IS3
     19            CALL TOTALS
     20      70    IF(IS4 .EQ. 0) GO TO 110
     21            IRC1=IS4
     22            CALL TOTALS
     23      C   NET REP COMPUTES YEARLY AND TOTAL COSTS FOR REPLACEMENT AND MAINT
     24      110   IF(IS5+IS6.EQ. 0) GO TO 130
     25            IRC1=IS5
     26            IRC2=IS6
     27            CALL NETREP
     28      C   ENERGY COMPUTES YEARLY AND TOTAL COSTS FOR EACH FUEL  TYPE
     29      130   IF(IS7.EQ. 0) GO TO 150
     30            IRC1=IS7
     31            CALL ENERGY
     32      C   TAXCST  COMPUTES TAX AND DEDUCTIONS  PROPERTY INTEREST ETC
     33      150   IF(IS9+IS8+IS12+IS14+IS15 .EQ. 3) GO TO 170
     34            IRC1=IS9
     35            IRC2=IS8
     36            IRC3=IS12
     37            IRC4=IS14
     38            IRC5=IS15
     39            CALL TAXCST
     40      C  IF THIS HVAC IS FOR COMMERCIAL USE  ADDDED INCOME IS COMPUTED
     41      C   DEPRECIATION IS ALSO COMPUTED
     42      170   IF(IS10+IS11+IS16+IS17 .EQ. 0) GO TO 190
     43            IRC1=IS10
     44            IRC2=IS11
     45            IRC3=IS16
     46            IRC4=IS17
     47            IF (ICOM.EQ. 1)CALL DEPINC
     48      190   IF(IS13 .EQ. 0) GO TO 210
     49            IRC1=IS13
     50            CALL SALVAG
     51      C   COMPUTE TOTAL PV AND AC
     52      210   IRC1=IS1
     53            CALL PVAC
     54            RETURN
     55      A21   WRITE(6,A21)
```

```
    56       601   FORMAT(5X,'ERROR IN SOLCST')
    57             RETURN
    58             END

@PRT.S A.SUNRIZ
```

SH*COST(1).SUNRIZ

```
      1             SUBROUTINE SUNRIZ(IPLTOP)
      2             COMMON/SEC/ISECT
      3             COMMON/XC/XCNT
      4             INTEGER XCNT
      5             COMMON/LAST/PRICEL
      6             COMMON/NAM/NAMES
      7             CHARACTER*63 NAMES
      8             DIMENSION CYLF(6)
      9             DO 8 I=1,ISECT-1
     10             READ(9'I,ERR=600) ISECT1,VARY,PCT,G,FLATE1,DISC1
     11            +,DNPMT,CSTPSM,TINCOM,PTAX,AREA,SHVAC,CHVAC,CYLF
     12            +,PSHAC,PCHAC
     13                 IF(MOD(I-1,40).NE.0) GO TO 15
     14                 CALL PAGES(99)
     15              WRITE(6,101) NAMES
     16             IF(IPLTOP.NE.1) WRITE(6,102) AREA
     17             IF(IPLTOP.NE.2) WRITE(6,103) DISC1
     18             IF(IPLTOP.NE.3) WRITE(6,104) FLATE1
     19             IF(IPLTOP.NE.4) WRITE(6,105) DNPMT
     20             IF(IPLTOP.NE.5) WRITE(6,106) PTAX
     21             IF(IPLTOP.NE.6) WRITE(6,107) TINCOM
     22             WRITE(6,108) CSTPSM
     23             IF(IPLTOP.EQ.1) WRITE(6,109)
     24             IF(IPLTOP.EQ.2) WRITE(6,110)
     25             IF(IPLTOP.EQ.3) WRITE(6,111)
     26             IF(IPLTOP.EQ.4) WRITE(6,112)
     27             IF(IPLTOP.EQ.5) WRITE(6,113)
     28             IF(IPLTOP.EQ.6) WRITE(6,114)
     29             WRITE(6,115)
     30       15    WRITE(6,100) ISECT1,VARY,PCT,G
     31            +,SHVAC,CHVAC,PSHAC,PCHAC,CYLF
     32       8     CONTINUE
     33             RETURN
     34       600   WRITE(6,601)
     35       601   FORMAT(5X,'ERROR IN SUNRIZ')
     36             RETURN
     37       101   FORMAT(T60,'SUMMARY',//,T55,A63,////)
     38       115   FORMAT(//,'  SECT',5X,'VARY  PERCNT    TOTAL',7X,'ACTUAL COST
     39            +PRESENT VAL COST',7X,'LIFE CYCLE COST SAVINGS',7X,'LIFE CYCLE FUEL
     40            + COST',/,'   NO',14X,'LOAD',5X,'LOAD    SHVAC',6X,'CHVAC    SHVAC',
     41            +  5X,'CHVAC    PRESENT',5X,'ACTUAL   ANNUAL   PRESENT    ACTUAL
     42            +ANNUAL',/,2X,8('-'),2X,8('-'),2X,6('-'),2X,7('-'),3X,8('-'),1X,
     43            +8('-'),2X,8('-'),1X,8('-'),3X,5(8('-'),2X),8('-'))
     44       100   FORMAT(1X,I5,2X,F8.1,2X,F5.2,'%',2X,F7.2,2X,2F9.2,1X,2F9.2,
     45            +2X,5(F9.2,1X),F9.2)
     46       102   FORMAT(T50,'COLLECTOR AREA          = ',F6.2,'GJ/YR')
     47       103   FORMAT(T50,'DISCOUNT RATE           = ',F6.2,'%')
     48       104   FORMAT(T50,'INFLATION RATE          = ',F6.2,'%')
     49       105   FORMAT(T50,'DOWNPAYMENT             = ',F6.2,'%')
     50       106   FORMAT(T50,'PROPERTY TAX RATE       = ',F6.2,'%')
     51       107   FORMAT(T50,'INCOME TAX RATE         = ',F6.2,'%')
     52       108   FORMAT(T50,'COST PER SQ. METER       $',F6.2)
     53       109   FORMAT(T46,'FOR THIS CASE, VARY IS THE COLECTOR AREA.  GJ/YR')
     54       110   FORMAT(T46,'FOR THIS CASE, VARY IS THE DISCOUNT RATE.  %')
     55       111   FORMAT(T46,'FOR THIS CASE, VARY IS THE INFLATION RATE.  %')
```

```
     56        112   FORMAT(T96,'FOR THIS CASE, VARY IS THE DOWNPAYMENT.  $')
     57        113   FORMAT(T96,'FOR THIS CASE, VARY IS THE PROPERTY TAX RATE.  $')
     58        114   FORMAT(T96,'FOR THIS CASE, VARY IS THE INCOME TAX RATE.  $')
     59              END

@PRT,S A.TAXCST
```

SH*COST(1).TAXCST

```
      SUBROUTINE TAXCST
      INCLUDE PARS
      COMMON /NFLT/ FLATE
      CALL SBLNK
      IF(IRC2.EQ.0) GO TO 10
C    COMPUTE PROPERTY TAX AND DEDUCTIONS
C    COST(1 THRU NYR)= ASSESSED VALUE OF HVAC IN YEAR J
C    COST(2)= PROPERTY TAX RATE
      READ(8'IRC2,ERR=600) ISUB,CODE,TITLE,COST,
     *FLCOST
10     IF(IRC2 .EQ. 0) GO TO 150
      READ(8'IRC1,ERR=600) ISA,ACODE,ATITLE,ACOST,
     *AFCOST
      IF (COST(2) .GT. 0.0 .AND. ACOST(2) .GT. 0.0 .AND.
     *ACOST(3) .GT. 0.0) GO TO 50
C       WRITE(6,300)
      GO TO 150
   50 DO 100 I=1,NYR
      IF(FLATE .EQ. 0.0)ACTUAL(I)=COST(2)*COST(1)
      IF(FLATE .NE. 0.0)ACTUAL(I)=COST(2)*FLCOST(I)
          TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
        TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
       ATOTAL(I)=ACOST(3)*TOTAL(I)
         ATOTAL(NYR+1)=ATOTAL(NYR+1)+ATOTAL(I)
      BCTUAL(I)=ACOST(3)*ACTUAL(I)
      ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
      BCTUAL(NYR+1)=BCTUAL(NYR+1)+BCTUAL(I)
  100   CONTINUE
        TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT
        ATOTAL(NYR+2)=ATOTAL(NYR+1)*ACFACT
      WRITE(8'IRC1,ERR=600) ISA,ACODE,ATITLE,
     *ACOST,AFCOST,ATOTAL,BCTUAL
      WRITE(8'IRC2,ERR=600) ISUB,CODE,TITLE,
     *COST,FLCOST,TOTAL,ACTUAL
  150 CALL SBLNK
      IF(IRC3.EQ. 0) GO TO 170
C    COMPUTE INSURANCE COSTS
      READ(8'IRC3,ERR=600) ISUB,CODE,TITLE,COST,
     *FLCOST
C    COST(1)=ANNUAL PREMIUMS  COST(2)=ANNUAL DAMAGE COST(3)=REIMBURSEMENTS
        TOTAL(NYR+2)=COST(1)*COST(2)-COST(3)
      DO 160 J=1,NYR
      ACTUAL(J)=TOTAL(NYR+2)
      ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(J)
  160 CONTINUE
        TOTAL(NYR+1)=TOTAL(NYR+2)/ACFACT
      WRITE(8'IRC3,ERR=600) ISUB,CODE,TITLE,COST,
     *FLCOST,TOTAL,ACTUAL
      CALL SBLNK
  170 IF(IRC4+IRC5.EQ. 0) GO TO 250
      READ(8'IRC4,ERR=600) ISUB,CODE,TITLE,COST,
     *FLCOST
      READ(8'IRC5,ERR=600) ISA,ACODE,ATITLE,ACOST,
     *AFCOST
```

```
      C      WRITE(6,310)
             GO TO 250
      C   COST(1) THRU COST(NYR) =OUTSTANDING MORTGAGE
      C    COST(2)= MARKET RATE
      C   COST(3)=INCOME TAX RATE
        199 DO 200 I=1,NYR
            ACTUAL(I)=COST(I)*COST(NYR+1)*COST(NYR+2)
            ACTUAL(NYR+1)=ACTUAL(NYR+1)+ACTUAL(I)
              TOTAL(I)=ACTUAL(I)/(1.0+DISC)**I
            BCTUAL(I)=ACOST(I)
            ATOTAL(I)=BCTUAL(I)/(1.0+DISC)**I
            BCTUAL(NYR+1)=BCTUAL(NYR+1)+BCTUAL(I)
            ATOTAL(NYR+1)=ATOTAL(NYR+1)+ATOTAL(I)
        200   TOTAL(NYR+1)=TOTAL(NYR+1)+TOTAL(I)
            ATOTAL(NYR+2)=ATOTAL(NYR+1)+ACFACT
            TOTAL(NYR+2)=TOTAL(NYR+1)+ACFACT
            WRITE(8'IRC4,ERR=600) ISUB,CODE,TITLE,COST,
           +FLCOST,TOTAL,ACTUAL
            WRITE(8'IRC5,ERR=600) ISA,ACODE,ATITLE,ACOST,
           +AFCOST,ATOTAL,BCTUAL
        250 RETURN
        600   WRITE(6,601)
        601   FORMAT(5X,'ERROR IN TAXCST')
              RETURN
      C 300 FORMAT(1H1,3X,'THE VALUE OF THE PROPERTY TAX RATE OR THE INCOME',
      C    +/,3X,'TAX RATE FOR PROPERTY TAX CALCULATIONS WAS ZERO. THE',//,3X,
      C    +'CALCULATIONS FOR PROPERTY TAX AND PROP TAX DEDUCTIONS WERE NOT',//,
      C    +3X,'PERFORMED',/)
      C 310 FORMAT(1H1,3X,'THE VALUE OF THE INCOME TAX RATE OR THE MARKET',//,
      C    +3X,'INTEREST RATE WAS ZERO. CALCULATIONS FOR DEDUCTIONS ON',//,3X,
      C    +'THE LOAN INTEREST WERE NOT PERFORMED.',//)
             END

@PRT,S A.TOTALS
```

```
SH*COST(1).TOTALS
      1              SUBROUTINE TOTALS
      2              INCLUDE PARS
      3        C    TOTALS ALL COST ACQ BLDG,AND SPACE VALUE GOES TO TOTALS
      4        1       CONTINUE
      5              IF(IRC1 .EQ.0) GO TO 100
      6              CALL SBLNK
      7              KAT1=6
      8              KAT2=IRC1
      9              READ(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     10             *FLCOST
     11               IF(ISUB(2) .GT. 0 .AND. COST(1) .LE.0) GO TO 20
     12               DO 10 J=1,NYR
     13          10     TOTAL(NYR+1)=TOTAL(NYR+1)+COST(J)
     14             ACTUAL(NYR+1)=TOTAL(NYR+1)
     15               GO TO 50
     16        C    SUM COSTS FOR EACH LEVEL
     17           20  DO 40 I=2,K1
     18                 IF(ISUB(I).LE.0)GOTO 40
     19                 ATOTAL(NYR+1)=0.0
     20              KAT1=17
     21              KAT2=ISUB(I)
     22                READ(8'ISUB(I),ERR=600) ISA,ACODE,ATITLE,
     23             *ACOST,AFCOST
     24                DO 30 J=1,NYR
     25           30     ATOTAL(NYR+1)=ATOTAL(NYR+1)+ACOST(J)
     26                BCTUAL(NYR+1)=ATOTAL(NYR+1)
     27              ATOTAL(NYR+3)=ATOTAL(NYR+1)*DWNPMT
     28              ATOTAL(NYR+2)=ATOTAL(NYR+1)*ACFACT*DWNPMT
     29              KAT1=35
     30              KAT2=ISUB(I)
     31              WRITE(8'ISUB(I),ERR=600) ISA,ACODE,ATITLE,
     32             *ACOST,AFCOST,ATOTAL,BCTUAL
     33              TOTAL(NYR+1)=TOTAL(NYR+1)+ATOTAL(NYR+1)
     34              ACTUAL(NYR+1)=ACTUAL(NYR+1)+BCTUAL(NYR+1)
     35           40  CONTINUE
     36            50 TOTAL(NYR+3)=TOTAL(NYR+1)*DWNPMT
     37               TOTAL(NYR+2)=TOTAL(NYR+1)*ACFACT*DWNPMT
     38               WRITE(8'IRC1,ERR=600) ISUB,CODE,TITLE,COST,
     39              *FLCOST,TOTAL,ACTUAL
     40        100    RETURN
     41        600    WRITE(5,601) KAT1,KAT2
     42        601    FORMAT(5X,'ERROR IN TOTALS',2I10)
     43                RETURN
     44        C       DEBUG TRACE,SUBTRACE,INIT
     45        C       AT 1
     46        C       TRACE ON
     47              END

$PRT,S A.TABLE
```

SH*COST(1).TABLE

```
      1             SUBROUTINE TABLE(I,J,PA,PB,PC,PD,PE,PF,NAME,L)
      2             COMMON/CITYS/NAMES(151)
      3             INTEGER AX(2,151)
      4             INTEGER BX(2,151),CX(2,151),DX(2,151),EX(2,151),FX(2,151)
      5             REAL L0(151),L
      6             DIMENSION A(2,151),B(2,151),C(2,151),
      7            *          D(2,151),E(2,151),F(2,151)
      8                  CHARACTER*63 NAMES,NAME
      9                  DATA (NAMES(I),I=1,10)/
     10            +  'ABILENE, TEXAS
     11            +','ALBANY, NEW YORK
     12            +','ALBUQUERQUE, NEW MEXICO
     13            +','AMARILLO, TEXAS
     14            +','AMES, IOWA
     15            +','AMHERST, MASSACHSETTS
     16            +','ANNAPOLIS, MARYLAND
     17            +','APALACHICOLA, FLORIDA
     18            +','ASHVILLE, NORTH CAROLINA
     19            +','ASTORIA, OREGON
     20            +'/
     21                  DATA (NAMES(I),I=11,20)/
     22            +  'ATLANTA, GEORGIA
     23            +','ATLANTIC CITY, NEW JERSEY
     24            +','BIG SPINGS, TEXAS
     25            +','BILLINGS, MONTANA
     26            +','BINGHAMPTON, NEW YORK
     27            +','BIRMINGHAM, ALABAMA
     28            +','BISMARK, NORTH DAKOTA
     29            +','BLUE HILL, MASSACHSETTS
     30            +','BOISE, IDAHO
     31            +','BOSTON, MASSACHSETTS
     32            +'/
     33                  DATA (NAMES(I),I=21,30)/
     34            +  'BOULDER, COLORADO
     35            +','BROWNSVILLE, TEXAS
     36            +','CAPE HATTERAS, NORTH CAROLINA
     37            +','CARIBOU, MAINE
     38            +','CHARLESTON, SOUTH CAROLINA
     39            +','CHAROLLE, NORTH CAROLINA
     40            +','CHATTANOOGA, TENNESSEE
     41            +','CHICAGO, ILLINOIS
     42            +','CLEAVELAND, OHIO
     43            +','COLUMBIA, MISSOURI
     44            +'/
     45                  DATA (NAMES(I),I=31,40)/
     46            +  'COLUMBUS, OHIO
     47            +','CORPUS CHRISTI, TEXAS
     48            +','CORVALLIS, OREGON
     49            +','DALLAS, TEXAS
     50            +','DAVIS, CALIFORNIA
     51            +','DAYTON, OHIO
     52            +','DENVER, COLORADO
     53            +','DES MOINES, IOWA
     54            +','DETROIT, MICHIGAN
     55            +','DODGE CITY, KANSAS
```

```
     +*/
          DATA (NAMES(I),I=41,50)/
     +  *DULUTH, MINNESOTA
     +*,*EAST LANSING, MICHIGAN
     +*,*EL PASO, TEXAS
     +*,*ELY, NEVADA
     +*,*FARGO, NORTH DAKOTA
     +*,*FORT SMITH, ARKANSAS
     +*,*FORT WAYNE, INDIANA
     +*,*FORT WORTH, TEXAS
     +*,*FRESNO, CALIFORNIA
     +*,*GAINSVILLE, FLORIDA
     +*/
          DATA (NAMES(I),I=51,60)/
     +  *GLASGOW, MONTANA
     +*,*GRAND JUNCTION, COLORADO
     +*,*GRAND LAKE, COLORADO
     +*,*GREAT FALLS, MONTANA
     +*,*GREEN BAY, WISCONSIN
     +*,*GREENSBORO, NORTH CAROLINA
     +*,*GREENVILLE-SPARTANBURG, NORTH CAROLINA
     +*,*GRIFFIN, GEORGIA
     +*,*HARTFORD, CONNECTICUT
     +*,*HOUSTON, TEXAS
     +*/
          DATA (NAMES(I),I=61,70)/
     +  *INDIANAPOLIS, INDIANA
     +*,*INYOKERN, CALIFORNIA
     +*,*ITHICA, NEW YORK
     +*,*JACKSON, MISSISSIPPI
     +*,*JACKSONVILLE, FLORIDA
     +*,*KANSAS CITY, MISSOURI
     +*,*KEY WEST, FLORIDA
     +*,*LAKE CHARLES, LOUISIANA
     +*,*LANDER, WYOMING
     +*,*LANSING, MICHIGAN
     +*/
          DATA (NAMES(I),I=71,80)/
     +  *LARAMIE, WEST VIRGINIA
     +*,*LAS VEGAS, NEVADA
     +*,*LEMONT ILLINOIS
     +*,*LEXINGTON, KENTUCKY
     +*,*LINCOLN, NEBRASKA
     +*,*LITTLE ROCK, ARKANSAS
     +*,*LOS ANGELES, CALIFORNIA
     +*,*LOUISVILLE, KENTUCKY
     +*,*LYNN, MASSACHSETTS
     +*,*MACON, GEORGIA
     +*/
          DATA (NAMES(I),I=81,90)/
     +  *MADISON, WISCONSIN
     +*,*MANHATTAN, KANSAS
     +*,*MEDFORD, OREGON
     +*,*MEMPHIS, TENNESSEE
     +*,*MIAMI, FLORIDA
```

```
     112            +*,*MILWAUKEE, WISCONSIN
     113            +*,*MINN.-ST. PAUL, MINNISOTA
     114            +*,*MT. WEATHER, WEST VIRGINIA
     115            +*,*NASHVILLE, TENNESSEE
     116            +*/
     117                 DATA (NAMES(I),I=91,100)/
     118            +  *NATICK, MASSACHSETTS
     119            +*,*NEW ORLEANS, LOUISIANA
     120            +*,*NEWPORT, RHODE ISLAND
     121            +*,*NEW YORK, NEW YORK
     122            +*,*NORFOLK, VIRGINIA
     123            +*,*NORTH OMAHA, NEBRASKA
     124            +*,*OAK RIDGE, TENNESSEE
     125            +*,*OKLAHOMA CITY, OKLAHOMA
     126            +*,*PAGE, OREGON
     127            +*,*PARKERSBURG, WEST VIRGINIA
     128            +*/
     129                 DATA (NAMES(I),I=101,110)/
     130            +  *PASADENA, CALIFORNIA
     131            +*,*PENSICOLA, FLORIDA
     132            +*,*PEORIA, ILLINOIS
     133            +*,*PHOENIX, ARIZONA
     134            +*,*PHILADELPHIA, PENNSYILVANIA
     135            +*,*PITTSBURGH, PENNSYLVANIA
     136            +*,*PORTELLO, IDAHO
     137            +*,*PORT AUTHOR, TEXAS
     138            +*,*PORTLAND, MAINE
     139            +*,*PORTLAND, OREGON
     140            +*/
     141                 DATA (NAMES(I),I=111,120)/
     142            +  *PROSSER, WEST VIRGINIA
     143            +*,*PUEBLO, COLORADO
     144            +*,*PULLMAN, WEST VIRGINIA
     145            +*,*PUT-IN-BAY, OHIO
     146            +*,*RALEIGH, NORTH CAROLINA
     147            +*,*RALEIGH-DURHAM, NORTH CAROLINA
     148            +*,*RAPID CITY, SOUTH DAKOTA
     149            +*,*RENO, NEVADA
     150            +*,*RICHLAND, WEST VIRGINIA
     151            +*,*RICHMOND, VIRGINIA
     152            +*/
     153                 DATA (NAMES(I),I=121,130)/
     154            +  *RIVERSIDE, CALIFORNIA
     155            +*,*ROCHCHESTER, NEW YORK
     156            +*,*SACRAMENTO, CALIFORNIA
     157            +*,*ST. CLOUD, MINNISOTA
     158            +*,*ST. LOUIS, MISSOURI
     159            +*,*SALT LAKE CITY, UTAH
     160            +*,*SAN ANTONIO, TEXAS
     161            +*,*SAN DIEGO, CALIFORNIA
     162            +*,*SAN FRANCISCO, CALIFORNIA
     163            +*,*SAN JOSE, CALIFORNIA
     164            +*/
     165                 DATA (NAMES(I),I=131,140)/
     166            +  *SANTA MARIA, CALIFORNIA
     167            +*,*SAVANNAH, GEORGIA
```

```
     +*,'SAULT ST. MARIE, MICHIGAN
     +*,'SCHENECEADY, NEW YORK
     +*,'SEATTLE, WASHINGTON
     +*,'SHREVEPORT, LOUISIANA
     +*,'SILVER HILL, MARYLAND
     +*,'SPOKANE, WASHINGTON
     +*,'STATE COLLEGE, PENNSYLVANIA
     +*,'STILLWATER, OKLAHOMA
     +*/
          DATA (NAMES(I),I=141,151)/
     +   'SUMMIT, MONTANA
     +*,'SYRACUSE, NEW YORK
     +*,'TALLAHASSEE, FLORIDA
     +*,'TAMPA, FLORIDA
     +*,'TRENTON, NEW JERSEY
     +*,'TUCSON, ARIZONA
     +*,'TULSA, OKLAHOMA
     +*,'TWIN FALLS, IDAHO
     +*,'WASHINGTON, DC
     +*,'WICHITA, KANSAS
     +*,'YUMA, ARIZONA
     +*/
      DATA (A(1,I),AX(1,I),I=1,60)/
     +.4966+-31,.1290+-01,.3606+-01,.3379+-01,.1365+-31,.9909+-02,
     1.1796+-01,.7609+-01,.2581+-01,.1676+-01,.3081+-01,.2042+-01,
     2.4591+-31,.1653+-31,.1042+-01,.3151+-01,.1353+-31,.1270+-01,
     3.1899+-01,.1211+-01,.2115+-01,.9212+-01,.4393+-01,.1068+-01,
     4.4831+-31,.3185+-31,.2007+-01,.1337+-01,.1363+-31,.1833+-01,
     5.1295+-01,.8908+-01,.1575+-01,.4056+-01,.3843+-01,.1440+-01,
     6.2968+-31,.1311+-31,.1106+-01,.2709+-01,.1327+-31,.9811+-02,
     7.5801+-01,.2227+-01,.9665+-02,.2512+-01,.1309+-01,.4531+-01,
     8.4164+-31,.7786+-31,.1512+-01,.2431+-01,.1623+-31,.1613+-01,
     9.1621+-01,.2068+-01,.3135+-01,.3709+-01,.1439+-01,.6308+-01/
      DATA (A(1,I),AX(1,I),I=61,120)/
     A.1271+-01,.7724+-01,.8769+-02,.3799+-01,.6995+-01,.1760+-01,
     1.1139+*00,.6319+-31,.2095+-01,.1098+-01,.1702+-31,.5589+-01,
     2.1310+-01,.2059+-01,.1748+-01,.2563+-01,.6201+-01,.1698+-01,
     3.1117+-31,.4991+-01,.1191+-01,.1829+-31,.1031+-31,.7655+-01,
     4.1063+*00,.8735+-01,.1162+-01,.1000+-01,.1696+-01,.2005+-01,
     5.1369+-31,.4925+-31,.1577+-01,.1147+-01,.2673+-31,.1683+-01,
     6.1960+-01,.3150+-01,.9119+-01,.1352+-01,.6335+-01,.6056+-01,
     7.1393+-31,.9168+-31,.1793+-01,.1535+-31,.1877+-31,.5613+-01,
     8.1954+-01,.1309+-01,.1951+-01,.2786+-01,.1649+-01,.1027+-01,
     9.2876+-31,.2760+-31,.1898+-01,.2868+-31,.1578+-01,.2233+-01/
      DATA (A(1,I),AX(1,I),I=121,151),(A(2,I),AX(2,I),I=1,29)/
     A.6842+-31,.1145+-31,.3831+-31,.1211+-31,.1799+-31,.1825+-01,
     1.6537+-01,.5838+-01,.3631+-01,.4341+-01,.4982+-01,.4986+-02,
     2.1159+-31,.8697+-32,.1905+-31,.3886+-31,.1995+-31,.1948+-01,
     3.1112+-01,.2739+-01,.1019+-01,.9797+-02,.6911+-01,.8726+-01,
     4.1896+-31,.8232+-31,.2949+-01,.1678+-31,.1804+-31,.2305+-01,
     5.1080+*00,.6934+-01,.1688+-01,.5337+-01,.4881+-01,.1837+-01,
     1.1399+-31,.2539+-31,.1192+*00,.3604+-31,.2193+-31,.9365+-31,
     2.2882+-01,.6539+-01,.2337+-01,.1380+-01,.4960+-01,.1867+-01,
     3.1737+-31,.2632+-31,.1691+-31,.1039+-01,.1438+*00,.6306+-31,
     4.1926+-01,.7034+-01,.8513+-01,.3825+-01,.1830+-01,.1356+-01/
      DATA (A(2,I),AX(2,I),I=30,60)/
```

```
224            5.2561•-01•,.1648•-01•,.1267•-01•,.2029•-01•,.5839•-01•,.5362•-01•
225            6.2002•-01•,.3584•-01•,.1813•-01•,.1956•-01•,.6893•-01•,.1398•-01•
226            7.1330•-01•,.8370•-01•,.3297•-01•,.1320•-01•,.3549•-01•,.1790•-01•
227            8.6511•-01•,.5837•-01•,.1184•+00•,.2138•-01•,.3995•-01•,.2300•-01•
228            9.2279•-01•,.1366•-01•,.3961•-01•,.8838•-01•,.5279•-01•,.1993•-01•
229            A.9176•-01•,.1701•-01•,.1188•-01•,.1157•-01•,.5790•-01•,.9650•-01•
230            1.2073•-01•,.1722•+00•,.9288•-01•,.3026•-01•,.1913•-01•,.2968•-01•
231            2.8061•-01•,.1885•-01•,.2899•-01•,.2941•-01•,.3609•-01•,.9270•-01•
232            3.2237•-01•,.1918•-01•,.4380•-01•,.1605•-01•,.2550•-01•,.2912•-01•
233            4.3749•-01•,.1611•-01•,.6817•-01•,.1573•-01•,.1369•-01•,.2288•-01/
234             DATA ((A(2,I),AX(2,I),I=9,15)/
235            5.2783•-01•,.1862•-01•,.7253•-01•,.2215•-01•,.1572•-01•,.3769•-01•
236            6.2351•-01•,.2668•-01•,.9993•-01•,.9999•-01•,.1797•-01•,.9371•-01•
237            7.8882•-01•,.1857•-01•,.1364•+00•,.2365•-01•,.2100•-01•,.9609•-01•
238            8.8113•-01•,.2028•-01•,.1675•-01•,.2605•-01•,.4013•-01•,.2203•-01•
239            9.1379•-01•,.4219•-01•,.3892•-01•,.2684•-01•,.8129•-01•,.2059•-01•
240            A.9122•-01•,.1035•+00•,.1593•-01•,.5397•-01•,.1672•-01•,.2976•-01•
241            1.2519•-01•,.9561•-01•,.8889•-01•,.5538•-01•,.6159•-01•,.7396•-01•
242            2.8216•-01•,.1998•-01•,.1134•-01•,.1178•-01•,.5555•-01•,.2767•-01•
243            3.1992•-01•,.1465•-01•,.3867•-01•,.1295•-01•,.1276•-01•,.9601•-01•
244            4.1331•+00•,.2508•-01•,.1212•+00•,.3462•-01•,.2255•-01•,.2695•-01•
245            5.3230•-01•,.1435•+00/
246             DATA ((B(1,I),BX(1,I),I=1,50)/
247            •-.1840•-02•,-.8575•-03•,-.1093•-02•,-.1107•-02•,-.8160•-03•
248            1-.9562•-03•,-.1183•-02•,-.3374•-02•,-.1252•-02•,-.1461•-02•
249            2-.1597•-02•,-.1060•-02•,-.1889•-02•,-.7642•-03•,-.9159•-03•
250            3-.1767•-02•,-.7344•-03•,-.8961•-03•,-.8183•-02•,-.9827•-03•
251            4-.8627•-03•,-.2870•-02•,-.1798•-02•,-.7104•-03•,-.2490•-02•
252            5-.1589•-02•,-.1986•-02•,-.9110•-03•,-.1391•-02•,-.1089•-02•
253            6-.1113•-02•,-.3511•-02•,-.1550•-02•,-.2037•-02•,-.2289•-02•
254            7-.1343•-02•,-.7773•-03•,-.8789•-03•,-.1325•-02•,-.9953•-03•
255            8-.7816•-03•,-.9807•-03•,-.1736•-02•,-.6523•-03•,-.7276•-02•
256            9-.1470•-02•,-.9779•-03•,-.1958•-02•,-.2102•-02•,-.2888•-02/
257             DATA (B(1,I),BX(1,I),I=51,100)/
258            4-.6820•-03•,-.9295•-03•,-.6358•-03•,-.8338•-03•,-.8381•-03•
259            5-.1335•-02•,-.1576•-02•,-.1804•-02•,-.8335•-03•,-.3375•-02•
260            0-.1398•-02•,-.1951•-02•,-.9620•-03•,-.2396•-02•,-.3125•-02•
261            1-.1066•-02•,-.6171•-03•,-.3320•-02•,-.6305•-03•,-.9649•-03•
262            2-.5775•-03•,-.1736•-02•,-.9135•-03•,-.1176•-02•,-.1398•-02•
263            3-.1553•-02•,-.1885•-02•,-.1738•-02•,-.1062•-02•,-.2194•-02•
264            4-.8735•-03•,-.9951•-03•,-.1411•-02•,-.1561•-02•,-.6332•-03•
265            5-.1819•-02•,-.8812•-03•,-.8019•-03•,-.9149•-03•,-.1489•-02•
266            6-.9331•-03•,-.3253•-02•,-.9538•-03•,-.1118•-02•,-.1412•-02•
267            7-.8829•-03•,-.1455•-02•,-.1301•-02•,-.9061•-03•,-.1242•-02/
268             DATA ((B(1,I),BX(1,I),I=101,150)/
269            8-.2183•-02•,-.3232•-02•,-.9927•-03•,-.2835•-02•,-.1089•-01•
270            9-.1082•-02•,-.9066•-03•,-.3261•-02•,-.7919•-03•,-.1427•-02•
271            7-.1251•-02•,-.8991•-03•,-.1116•-02•,-.9663•-03•,-.1366•-02•
272            8-.1377•-02•,-.7423•-03•,-.8038•-03•,-.1399•-02•,-.1306•-02•
273            1-.2388•-02•,-.9825•-03•,-.1990•-02•,-.6769•-03•,-.1169•-02•
274            2-.1339•-02•,-.3931•-02•,-.1938•-02•,-.5952•-03•,-.2223•-02•
275            3-.1060•-02•,-.2568•-02•,-.8603•-03•,-.8778•-03•,-.1564•-02•
276            4-.9192•-02•,-.1212•-02•,-.1923•-02•,-.1033•-02•,-.1396•-02•
277            5-.8722•-03•,-.9759•-03•,-.2808•-02•,-.1607•-02•,-.1065•-02•
278            6-.2683•-02•,-.1332•-02•,-.1214•-02•,-.1217•-02•,-.1123•-02/
279             DATA ((C(1,I),CX(1,I),C(2,I),CX(2,I),I=1,83)/
```

```
280          0-.2809E-02,-.2886E-02,-.1492E-02,-.1726E-02,-.1827E-02,
281          1-.1346E-02,-.1536E-02,-.1899E-02,-.4835E-02,-.1959E-02,
282          2-.1269E-02,-.2570E-02,-.1863E-02,-.3010E-02,-.1349E-02,
283          3-.1983E-02,-.2880E-02,-.1162E-02,-.1594E-02,-.1848E-02,
284          4-.1628E-02,-.1465E-02,-.4746E-02,-.2893E-02,-.1189E-02,
285          5-.3768E-02,-.5211E-02,-.2811E-02,-.1991E-02,-.1665E-02,
286          6-.1796E-02,-.1715E-02,-.5557E-02,-.2392E-02,-.3311E-02,
287          7-.3536E-02,-.1667E-02,-.1263E-02,-.1943E-02,-.1584E-02,
288          8-.1639E-02,-.1211E-02,-.1478E-02,-.2676E-02,-.1115E-02,
289          9-.1199E-02,-.2463E-02,-.1618E-02,-.3142E-02,-.3955E-02/
290           DATA (B(2,I),BX(2,I),I=0,99)/
291          A-.9673E-02,-.1157E-02,-.[illegible]995E-02,-.13[illegible]02,-.1336E-02,
292          B-.1333E-02,-.2189E-02,-.[illegible]2580E-02,-.2808E-02,-.1380E-02,
293          0-.5259E-02,-.[illegible]1711E-02,-.3377E-02,-.1985E-02,-.3999E-02,
294          1-.9859E-02,-.[illegible]1756E-02,-.9491E-03,-.5297E-02,-.1029E-02,
295          2-.1969E-02,-.[illegible]1938E-02,-.2795E-02,-.1999E-02,-.1998E-02,
296          3-.1558E-02,-.[illegible]2611E-02,-.3031E-02,-.1913E-02,-.1697E-02,
297          4-.1959E-02,-.1339E-02,-.1668E-02,-.2315E-02,-.2621E-02,
298          5-.9139E-03,-.2922E-02,-.1496E-02,-.1294E-02,-.1538E-02,
299          6-.2292E-02,-.[illegible]613E-02,-.5185E-02,-.162[illegible]02,-.1895E-02,
300          7-.2305E-02,-.1368E-02,-.2280E-02,-.212[illegible]02,-.1432E-02/
301           DATA (B(2,I),BX(2,I),I=100,151)/
302          8-.1897E-02,-.3622E-02,-.5027E-02,-.1577E-02,-.4556E-02,
303          9-.[illegible]789E-02,-.1768E-02,-.1534E-02,-.5013E-02,-.1379E-02,
304          A-.2167E-02,-.2053E-02,-.1434E-02,-.1836E-02,-.1624E-02,
305          B-.2133E-02,-.2243E-02,-.1203E-02,-.1439E-02,-.2164E-02,
306          0-.2131E-02,-.3570E-02,-.1543E-02,-.3353E-02,-.1107E-02,
307          1-.1876E-02,-.1669E-02,-.9681E-02,-.3333E-02,-.1432E-02,
308          2-.3665E-02,-.1834E-02,-.3963E-02,-.1325E-02,-.1400E-02,
309          3-.2985E-02,-.3563E-02,-.1985E-02,-.1665E-02,-.1595E-02,
310          4-.2237E-02,-.1307E-02,-.1519E-02,-.4361E-02,-.1951E-02,
311          5-.1758E-02,-.4106E-02,-.2179E-02,-.1688E-02,-.2018E-02,
312          6-.1797E-02,-.4298E-02/
313           DATA (C(1,I),CX(1,I),I=1,50)/
314          0-.1183E-04,-.8409E-05,.7717E-05,.3930E-06,-.1568E-04,
315          1.9698E-05,.3196E-05,-.7028E-05,-.2569E-04,-.4538E-05,
316          2.2226E-05,.2481E-05,-.6195E-05,.1416E-05,-.2741E-05,
317          3.1939E-05,-.2395E-04,-.1329E-04,-.1559E-04,-.1134E-04,
318          4-.4025E-06,-.9029E-04,-.1209E-04,-.1758E-04,-.4184E-04,
319          5-.2513E-06,-.8623E-05,-.1135E-04,-.5355E-05,-.1486E-05,
320          6-.1323E-04,-.1428E-04,-.3636E-05,.1729E-04,-.2977E-04,
321          7-.1895E-06,-.8975E-05,-.8543E-05,-.1295E-04,-.6475E-05,
322          8-.1318E-04,.1185E-04,-.8989E-06,.3629E-05,.1879E-05,
323          9-.9337E-05,-.3275E-05,-.2241E-04,-.1202E-04,.6966E-05/
324           DATA (C(1,I),CX(1,I),I=51,100)/
325          A-.7019E-05,-.2189E-04,-.5638E-06,-.1366E-04,-.3193E-05,
326          B-.4384E-05,-.1850E-05,.3921E-05,-.2677E-04,-.3746E-04,
327          1-.1625E-04,-.3133E-04,.1258E-04,-.1349E-04,-.2819E-04,
328          2-.6036E-05,-.6758E-05,-.2870E-04,.2481E-05,.1361E-05,
329          3-.8726E-05,-.4749E-05,-.7720E-06,-.8363E-07,-.2425E-04,
330          4-.1259E-04,-.1011E-04,0.1778E-04,.1019E-04,-.1041E-04,
331          5-.1378E-04,-.4470E-06,-.7568E-06,-.9339E-05,-.6621E-04,
332          6.3879E-05,-.5096E-05,.3937E-05,-.2144E-04,-.2204E-04,
333          7-.1813E-04,.4851E-05,-.9771E-05,-.4983E-07,-.5598E-05,
334          8-.1858E-04,-.9793E-05,-.1767E-[illegible],-.7686E-05,-.1619E-04/
335           DATA (C(1,I),CX(1,I),I=151,151)/
```

```
     9+.9886,-05,-.2633,-04,-.3485,-05,-.1515,-04,-.1802,-04,
     A-.7393,-05,-.2662,-05,-.3328,-04,-.1293,-04,-.2526,-05,
     B-.8884,-05,-.2233,-05,-.1322,-04,+.3953,-05,-.1987,-04,
     0-.2577,-04,-.1418,-04,+.1878,-05,-.2367,-05,+.8345,-06,
     1-.1311,-04,-.9870,-05,+.3334,-06,-.1717,-04,-.1001,-04,
     2-.2000,-04,-.2100,-04,-.9571,-05,-.2176,-04,-.3513,-04,
     3+.9838,-04,-.2125,-04,-.2355,-04,-.1988,-05,+.3860,-04,
     4+.1245,-04,-.1058,-04,-.1898,-04,-.8899,-05,-.7877,-05,
     5-.1578,-04,-.4789,-05,-.1310,-04,-.2516,-04,-.1095,-04,
     6+.1085,-04,-.6363,-05,-.3673,-04,+.3038,-06,-.2002,-04/
      DATA C(1,151),CX(1,151),(C(2,I),CX(2,I),I=1,49)/
     0-.3479,-04,-.1067,-04,+.1677,-05,+.9273,-05,-.2434,-04,
     1-.9237,-05,+.2097,-04,-.1855,-04,-.2252,-04,-.1561,-04,
     2+.1366,-04,-.1586,-04,-.2738,-04,-.2521,-04,-.3504,-04,
     3+.8170,-05,-.1468,-04,-.1103,-04,-.7208,-05,-.1921,-04,
     4-.9357,-05,-.3371,-04,-.1044,-03,-.2277,-04,-.1851,-04,
     5-.5013,-04,-.1226,-04,-.7688,-04,-.9007,-05,+.3060,-04,
     6-.1503,-04,+.1428,-04,-.7684,-04,+.8256,-04,+.1115,-04,
     7-.9224,-05,+.2533,-05,-.1264,-04,-.2342,-05,+.1403,-04,
     8-.4109,-05,+.1158,-04,-.1894,-04,+.1637,-05,-.2904,-04,
     9-.8666,-05,+.7699,-06,-.1002,-04,-.1178,-04,-.2235,-04/
      DATA (C(2,I),CX(2,I),I=50,99)/
     A-.3896,-04,-.1764,-04,-.1922,-04,-.3201,-04,-.3697,-04,
     B-.5262,-06,-.1321,-04,-.1488,-04,-.1163,-04,-.2164,-04,
     C-.3156,-04,+.1313,-04,+.3358,-05,+.2937,-04,-.2322,-05,
     0-.3612,-04,-.1213,-04,-.1211,-03,-.3442,-04,-.2032,-04,
     1+.1695,-05,-.3047,-04,-.6536,-05,-.9824,-05,-.2052,-04,
     2-.6636,-05,-.2994,-05,-.1366,-04,+.1516,-04,+.2417,-04,
     3-.1195,-04,-.6051,-05,-.1623,-04,+.1713,-05,+.2799,-05,
     4-.9214,-04,-.7205,-05,-.4001,-05,-.4812,-05,-.2237,-04,
     5+.3938,-04,-.1475,-04,-.1234,-04,-.3569,-04,-.1435,-04,
     6-.7795,-05,-.2321,-05,+.6036,-05,-.7578,-05,-.6852,-05/
      DATA (C(2,I),CX(2,I),I=100,151)/
     7+.2440,-04,-.1810,-04,-.4162,-04,+.5389,-06,-.3700,-04,
     8-.1805,-04,-.8224,-05,-.1999,-04,-.5533,-04,-.3394,-04,
     9+.5456,-04,-.8086,-05,-.4517,-05,-.1255,-04,-.6055,-07,
     A-.1597,-04,-.5223,-05,-.2196,-04,-.3398,-04,+.3932,-04,
     B-.1170,-04,-.3800,-04,-.3620,-05,-.3239,-04,-.7041,-05,
     1-.9113,-05,-.2287,-04,-.3063,-04,-.6875,-04,-.1230,-03,
     2-.5180,-04,-.4302,-06,-.4932,-04,+.5634,-05,+.1934,-04,
     3+.8812,-04,+.8560,-05,-.1413,-04,-.2136,-04,+.1692,-04,
     4-.4697,-06,+.1811,-04,+.2045,-04,-.1218,-04,+.8175,-04,
     5-.1406,-04,-.2523,-04,-.5700,-05,-.8680,-05,-.1338,-04,
     6-.1194,-04,-.6694,-04/
      DATA (D(1,I),DX(1,I),I=1,50)/
     +-.2183,-03,-.2767,-04,-.5499,-04,-.5661,-04,-.1860,-04,
     1-.1346,-04,-.3538,-04,-.5837,-03,-.6739,-04,-.4747,-04,
     2-.1143,-03,-.3439,-04,-.2482,-03,-.1810,-04,-.1420,-04,
     3-.1353,-04,-.1884,-04,-.1548,-04,-.5291,-04,-.1799,-04,
     4-.1087,-04,-.5324,-03,-.2173,-03,-.1325,-04,-.2980,-03,
     5-.1192,-03,-.7670,-04,-.2323,-04,-.1765,-04,-.3739,-04,
     6-.2266,-04,-.9637,-03,-.4408,-04,-.2117,-03,-.2501,-03,
     7-.2579,-04,+.1612,-04,-.1948,-04,-.1741,-04,-.4465,-04,
     8-.1425,-04,-.1329,-04,-.3385,-03,+.1064,-04,-.1076,-04,
     9-.7833,-04,-.2193,-04,-.2527,-03,-.2856,-03,-.3927,-03/
      DATA (D(1,I),DX(1,I),I=51,100)/
```

```
392             A-.1633,-04,-.4009,-04,-.1668,-05,-.2175,-04,-.1284,-04,
393             B-.6843,-04,-.1216,-03,-.1749,-03,-.1676,-04,-.6251,-03,
394             0-.2291,-04,-.7153,-03,-.1025,-04,-.2083,-03,-.4292,-03,
395             1-.3818,-04,+.5536,-02,-.5799,-03,+.8686,-05,-.1359,-04,
396             2+.1201,-04,-.3357,-03,-.1742,-04,-.5421,-04,-.2859,-04,
397             3-.8581,-04,+.1730,-03,-.3650,-04,-.1792,-04,-.2612,-03,
398             4-.1744,-04,-.3213,-04,-.5986,-04,-.9404,-04,+.4396,-02,
399             5-.2442,-03,-.1731,-04,-.1143,-04,-.2572,-04,-.5739,-04,
400             6-.2141,-04,-.3375,-03,-.1936,-04,-.1869,-04,-.7837,-04,
401             7-.2273,-04,-.5484,-04,-.9237,-04,-.1121,-04,-.2626,-04/
402              DATA (D(1,I),DX(1,I),I=101,150)/
403             8-.8437,-04,-.5399,-03,-.9236,-04,-.1054,-02,-.3914,-04,
404             9-.2941,-04,-.3913,-04,-.4921,-03,-.1613,-04,-.2653,-04,
405             A-.6534,-04,-.6087,-05,-.9260,-04,-.1978,-04,-.6752,-04,
406             B-.8777,-04,-.1577,-04,-.2479,-04,-.4369,-04,-.5848,-04,
407             1-.2411,-03,-.1689,-04,-.2394,-03,-.1223,-04,-.4371,-04,
408             2-.4391,-04,-.5690,-03,+.2662,-03,-.2744,-03,-.2545,-3,
409             3+.3572,-03,-.3383,-03,-.2985,-04,-.8643,-05,-.3490,-04,
410             4-.2120,-03,-.3563,-04,-.3320,-04,-.1694,-04,-.8297,-04,
411             5-.1566,-04,-.1307,-04,-.3232,-03,+.1457,-07,-.3488,-04,
412             6-.7821,-03,-.6403,-04,-.9280,-04,-.4018,-04,-.5336,-04/
413              DATA D(1,151),DX(1,151),(D(2,I),DX(2,I),I=1,49)/
414             7-.1033,-02,-.5819,-03,-.4632,-04,-.2563,-03,-.2370,-03,
415             1-.9797,-04,-.2753,-04,-.9530,-04,-.1765,-04,-.1798,-03,
416             2-.8222,-04,-.2960,-03,-.1147,-03,-.6278,-03,-.6820,-04,
417             3-.3035,-04,-.3237,-03,-.4728,-04,-.4165,-04,-.1144,-03,
418             4-.4022,-04,-.8644,-04,-.2082,-02,-.5616,-03,-.2993,-04,
419             5-.7688,-03,-.3014,-03,-.1867,-03,-.4964,-04,-.2943,-04,
420             6-.9998,-04,-.4560,-04,-.2551,-02,-.7259,-04,-.5351,-03,
421             7-.5134,-03,-.6474,-04,-.6683,-04,-.5127,-04,-.3485,-04,
422             8-.1504,-03,-.2791,-04,-.2770,-04,-.9013,-03,-.6266,-04,
423             9-.2729,-04,-.1997,-03,-.5179,-04,-.6303,-03,-.5944,-03/
424              DATA (D(2,I),DX(2,I),I=50,99)/
425             A-.1585,-02,-.5703,-04,-.1312,-03,-.4162,-04,-.7382,-04,
426             B-.2931,-04,-.1771,-03,-.3026,-03,-.4001,-03,-.5041,-04,
427             0-.1486,-02,-.4820,-04,-.1659,-02,-.2093,-04,-.4824,-03,
428             1-.1305,-02,-.9346,-04,+.1165,-01,-.1436,-02,-.4779,-04,
429             2-.3167,-04,-.3437,-04,-.8711,-03,-.4920,-04,-.1357,-03,
430             3-.8058,-04,-.2107,-03,-.2531,-03,-.8193,-04,-.3601,-04,
431             4-.6398,-03,-.3888,-04,-.9298,-04,-.1348,-03,-.2284,-03,
432             5+.8851,-02,-.6407,-03,-.3980,-04,-.2944,-04,-.6964,-04,
433             6-.1316,-03,-.5083,-04,-.8735,-03,-.6768,-04,-.3897,-04,
434             7-.2069,-03,-.6621,-04,-.1192,-03,-.2554,-03,-.1362,-03/
435              DATA (D(2,I),DX(2,I),I=100,151)/
436             8-.5360,-04,-.6393,-03,-.1317,-02,-.5561,-04,-.2786,-02,
437             9-.9371,-04,-.7026,-04,-.9769,-04,-.1126,-02,-.5767,-04,
438             A-.4615,-04,-.1210,-03,-.1047,-03,-.8249,-04,-.3125,-04,
439             B-.2154,-03,-.2132,-03,-.6733,-04,-.1397,-03,-.7612,-04,
440             0-.1491,-03,-.9571,-03,-.3837,-04,-.5086,-03,-.3489,-04,
441             1-.9819,-04,-.1035,-03,-.1426,-02,-.1561,-03,+.1030,-03,
442             2-.5832,-03,+.2540,-03,-.8316,-03,-.3536,-04,-.1838,-04,
443             3-.5538,-04,-.5962,-03,-.1159,-03,-.6525,-04,-.3466,-04,
444             4-.2195,-03,-.2713,-04,-.2565,-04,-.1124,-02,-.2129,-02,
445             5-.9214,-04,-.2095,-02,-.1752,-03,-.8239,-04,-.9951,-04,
446             6-.1402,-03,-.3288,-02/
447              DATA (D(1,I),DX(1,I),I=1,50)/
```

```
448            +.8983,-05,+.1840,-05,+.6823,-05,+.5993,-05,+.5386,-06,
449            1+.7923,-05,+.2576,-05,+.2555,-04,+.1655,-05,+.7418,-05,
450            2+.6885,-05,-.8035,-06,+.1253,-04,-.1888,-05,+.2602,-05,
451            3+.1059,-04,+.4151,-06,+.1219,-05,+.4178,-05,+.2343,-05,
452            4-.6477,-05,-.9016,-04,+.8177,-05,+.5207,-06,+.2372,-04,
453            5+.6378,-05,+.6511,-05,+.1533,-05,+.4133,-05,+.2360,-05,
454            6+.3895,-05,+.4528,-04,+.8938,-05,+.1439,-04,+.2443,-04,
455            7+.2832,-05,-.1010,-04,+.1833,-05,+.3296,-05,-.3231,-05,
456            8+.1457,-05,+.2903,-05,+.7476,-05,-.9039,-05,+.1337,-05,
457            9+.5318,-05,+.2795,-05,+.1139,-04,+.2178,-04,-.1859,-04/
458             DATA (E(1,I),EX(1,I),I=51,100)/
459            A-.1202,-05,-.1973,-05,-.3843,-05,-.7685,-06,+.1655,-05,
460            B+.9631,-05,+.7361,-05,+.1389,-04,-.7139,-06,+.5670,-04,
461            0+.3661,-05,+.1489,-04,+.3078,-05,-.1700,-04,+.2711,-04,
462            1+.2328,-05,-.9855,-03,+.5137,-04,-.7579,-05,+.2272,-05,
463            2-.7029,-05,+.8223,-05,+.1600,-05,+.3505,-05,+.1413,-05,
464            3+.6717,-05,-.8229,-04,+.4452,-05,+.3713,-05,+.1739,-04,
465            4+.1271,-05,+.8497,-06,+.8448,-05,+.7142,-05,+.8540,-03,
466            5+.8673,-05,+.2122,-05,+.1693,-05,-.1393,-06,+.6705,-05,
467            6+.1298,-05,+.3818,-04,-.1164,-06,+.3847,-05,+.3832,-05,
468            7+.3932,-06,+.6848,-05,+.9336,-06,-.9528,-05,+.8875,-05/
469             DATA (E(1,I),EX(1,I),I=101,150)/
470            8-.4753,-04,+.5018,-04,+.2620,-05,+.3509,-04,+.1993,-05,
471            9+.3001,-05,+.1195,-05,+.4931,-04,-.8572,-06,+.6965,-05,
472            A+.7015,-05,-.8690,-05,+.4293,-05,+.2873,-05,+.6995,-05,
473            B+.2988,-05,-.3614,-05,-.9975,-05,+.7697,-05,+.3931,-05,
474            1-.1451,-04,+.2633,-05,+.1824,-04,-.3299,-06,-.3451,-05,
475            2+.2929,-05,+.4095,-04,-.3598,-04,-.9152,-04,+.1152,-04,
476            3-.8923,-04,+.2738,-04,+.1553,-05,+.2339,-05,+.9232,-05,
477            4+.1840,-04,+.3111,-05,+.3057,-05,+.3288,-05,+.2972,-05,
478            5+.2483,-05,+.3090,-05,+.1563,-04,-.4250,-03,+.1555,-05,
479            6+.3923,-04,+.3052,-05,+.2346,-05,+.3359,-05,+.1384,-05/
480             DATA E(1,151),EX(1,151),(E(2,I),EX(2,I),I=1,49)/
481            7-.3144,-04,+.2848,-04,+.6591,-05,-.8658,-05,+.4112,-05,
482            1+.3536,-05,+.7467,-05,+.9920,-05,+.8529,-04,+.7663,-05,
483            2+.1840,-04,+.2145,-04,+.6069,-05,+.3471,-04,+.6992,-06,
484            3+.7057,-05,+.3089,-04,+.2869,-05,+.5959,-06,+.1203,-04,
485            4+.7730,-05,-.7214,-05,-.1055,-03,+.2853,-04,+.3104,-05,
486            5+.5321,-04,+.2053,-04,+.1961,-04,+.5894,-05,+.9385,-05,
487            6+.9150,-05,+.9797,-05,+.1412,-03,+.1982,-04,+.4402,-04,
488            7+.5811,-04,+.8789,-05,-.1598,-04,+.6178,-05,+.8232,-05,
489            8-.1186,-05,+.4314,-05,+.7219,-05,+.1996,-04,-.1431,-04,
490            9+.4271,-05,+.2043,-04,+.8216,-05,+.3722,-04,+.5846,-04/
491             DATA (E(2,I),EX(2,I),I=50,99)/
492            A+.1683,-04,+.9862,-06,-.5737,-06,-.5681,-05,+.1753,-05,
493            B+.5365,-05,+.1359,-04,+.2277,-04,+.3128,-04,+.2150,-05,
494            0+.1421,-03,+.7662,-05,+.3374,-04,+.7443,-05,+.8891,-04,
495            1+.8287,-04,+.9012,-05,-.2193,-02,+.1341,-03,-.1321,-04,
496            2+.6693,-05,-.3912,-05,+.2470,-04,+.6013,-05,+.1315,-04,
497            3+.5251,-05,+.2418,-04,-.1508,-03,+.1209,-04,+.9647,-05,
498            4+.5053,-04,+.4875,-05,+.6164,-05,+.1986,-04,+.2484,-04,
499            5-.1897,-02,+.2812,-04,+.6439,-05,+.5043,-05,+.3689,-05,
500            6+.1917,-04,+.5852,-05,+.1050,-03,+.4729,-04,+.1107,-04,
501            7+.1489,-04,+.2510,-05,+.1773,-04,+.8826,-05,-.1676,-04/
502             DATA (E(2,I),EX(2,I),I=100,151)/
503            8+.1208,-04,-.5623,-04,+.1258,-03,+.7852,-05,+.1208,-03,
```

```
     9+.8349,-05,+.9540,-05,+.5303,-05,+.1179,-06,+.2458,-05,
     A+.1539,-04,+.1696,-04,-.1211,-04,+.1153,-04,+.8583,-05,
     B+.6481,-05,+.1320,-04,-.4484,-05,-.1461,-04,+.1718,-04,
     0+.1384,-04,-.1430,-04,+.7561,-05,+.5391,-04,+.1467,-05,
     1+.1125,-04,+.8214,-05,+.1023,-03,+.1572,-33,+.1486,-03,
     2+.4362,-04,-.1641,-03,-.6800,-04,+.5131,-05,+.6377,-05,
     3+.2091,-04,+.5176,-04,+.1109,-04,+.8881,-05,+.8351,-05,
     4+.1446,-04,+.5500,-05,+.7699,-05,+.5357,-04,-.8396,-03,
     5+.7348,-05,+.9134,-04,+.1381,-04,+.8698,-05,+.1208,-04,
     6+.6735,-05,-.3523,-04/
      DATA (F(1,I),FX(1,I),I=1,60)/
     +.9167,-06,.5586,-03,.3106,-06,.3266,-06,.6468,-06,.2389,-06,
     1.4951,-06,.1649,-05,.1151,-05,.6108,-06,.5391,-06,.4916,-06,
     2.5796,-06,.3481,-06,.2757,-06,.5159,-06,.5960,-06,.6075,-06,
     3.4383,-06,.6620,-06,.7299,-06,.9504,-05,.9872,-06,.4306,-06,
     4.1421,-06,.5613,-06,.5872,-06,.5893,-06,.2239,-06,.4012,-06,
     5.3901,-06,.3040,-05,.1260,-06,.1665,-06,.7077,-06,.3508,-06,
     6.8112,-06,.3895,-06,.3889,-06,.6065,-06,.3508,-06,.1804,-06,
     7.6266,-06,.4511,-06,.1943,-06,.6873,-06,.3278,-06,.1561,-05,
     8.4311,-06,.2434,-05,.2754,-06,.8810,-06,.3592,-06,.4739,-06,
     9.2857,-06,.6974,-06,.5675,-06,.4482,-06,.8649,-06,.9180,-06/
      DATA (F(1,I),FX(1,I),I=61,120)/
     1.4132,-06,.2547,-05,.1827,-06,.3402,-06,.2454,-05,.4190,-06,
     2.7215,-04,.9146,-06,.2571,-06,.3034,-06,.6612,-06,.8465,-06,
     3.3725,-06,.3626,-06,.6812,-06,.6800,-06,.5314,-05,.5495,-06,
     4.2698,-06,.7982,-06,.4761,-06,.4629,-06,.1238,-06,.6927,-06,
     5.3815,-04,.4500,-04,.3079,-06,.2153,-06,.9897,-06,.6980,-06,
     6.6628,-06,.5379,-06,.4833,-06,.4511,-06,.7583,-06,.6574,-06,
     7.4731,-06,.1154,-05,.7016,-06,.4592,-06,.1860,-05,.5761,-06,
     8.3273,-06,.1544,-05,.8763,-06,.4528,-06,.3686,-06,.7508,-06,
     9.6340,-06,.3795,-06,.2254,-06,.4999,-06,.4660,-06,.3299,-06,
     0.1945,-05,.1370,-05,.6544,-06,.4873,-06,.1681,-37,.5014,-06/
      DATA (F(1,I),FX(1,I),I=121,151),(F(2,I),FX(2,I),I=1,29)/
     1.9672,-06,.3335,-06,.4134,-06,.5187,-06,.4813,-06,.6508,-06,
     2.9328,-06,.6825,-05,.5872,-05,.2601,-05,.8805,-06,.8887,-06,
     3.5563,-06,.3224,-06,-.2137,-6,.2674,-06,.6291,-06,.4496,-06,
     4.3594,-06,.7149,-06,.2888,-04,.2905,-04,.1890,-05,.1993,-04,
     5.7969,-06,-.2742,-6,.6694,-06,.9722,-06,.5973,-06,.8987,-06,
     0.6555,-05,.1802,-05,.6170,-04,.1811,-05,.2238,-05,.9341,-06,
     1.3757,-06,.1441,-05,.3735,-05,.1740,-05,.6129,-06,.1812,-05,
     2.2110,-05,.2182,-05,.1750,-05,.4277,-06,.1527,-05,.7172,-06,
     3.1364,-05,.8673,-06,.9930,-06,.2856,-05,.3316,-04,.2118,-05,
     4.9159,-06,.2767,-05,.1537,-05,.1238,-05,.9241,-06,.1451,-06/
      DATA (F(2,I),FX(2,I),I=30,89)/
     5.1156,-05,.4305,-06,.9736,-05,-.3130,-6,.1032,-05,.3262,-06,
     6.7161,-06,.2089,-05,.6655,-06,.3859,-06,.1334,-05,.3775,-06,
     7.3948,-06,.1364,-05,.2039,-05,.4892,-06,.1147,-05,.7147,-06,
     8.1998,-05,.8361,-06,.9753,-05,.9567,-06,.1438,-05,.1709,-05,
     9.1630,-05,.5232,-06,.1576,-05,.1648,-05,.1598,-05,.1557,-05,
     0.7918,-06,.4642,-06,.3657,-06,.2279,-06,.1056,-05,.4995,-05,
     1.1022,-05,.1018,-03,.1988,-05,.1622,-05,.5723,-06,.1986,-05,
     2.1730,-05,.9197,-06,.1089,-05,.1018,-05,.1267,-05,.1218,-04,
     3.6736,-06,.3909,-06,.1469,-05,.6774,-06,.1304,-05,-.1316,-6,
     4.9286,-06,.8552,-04,.1615,-05,.5793,-06,.5209,-06,.1587,-05/
      DATA (F(2,I),FX(2,I),I=90,151)/
     5.9777,-06,.1241,-05,.1237,-05,.1987,-05,.6444,-06,.1227,-05,
```

```
     6.8479,-06,.7333,-06,.1479,-05,.9811,-06,.9313,-06,.7618,-05,
     7.1208,-05,.6442,-06,.3880,-05,.1380,-05,.9243,-06,.1087,-05,
     8.1647,-05,.1597,-05,.1139,-06,.1588,-06,.1291,-05,.6351,-06,
     9.5069,-06,.2234,-05,.1659,-05,.1562,-05,.2332,-05,-.5540,-6,
     0.1993,-05,.6730,-05,.5991,-06,.1982,-05,.7639,-06,.1001,-05,
     1.1340,-05,.2609,-05,.2115,-04,.1873,-04,.5070,-05,.8281,-05,
     2.2732,-05,.5376,-05,.4373,-06,-.6951,-6,.8137,-06,.1910,-05,
     3.8182,-06,.4378,-06,.1367,-05,.2597,-06,.3332,-06,.2813,-05,
     4.1678,-04,.1396,-05,.1884,-05,.1420,-05,.8826,-06,.1366,-05,
     5.1391,-05,.1675,-04/
      DATA LD/63.07,129.69,89.23,87.53,128.61,124.75,93.21,92.82,98.37,
     1104.83,70.62,95.47,62.78,135.46,135.78,66.71,163.13,121.50,112.82,
     211D.1D,1DB.64,32.59,64.95,178.38,54.1D,72.52,76.39,117.77,118.19,
     3100.96,110.51,36.79,97.97,58.69,61.39,110.21,116.08,126.83,112.31,
     4103.96,174.2D,129.85,64.87,142.74,166.66,74.36,119.08,59.88,63.09,
     539.29,162.39,110.18,190.47,143.01,148.42,81.97,71.67,66.03,121.23,
     699.78,139.21,59.34,132.15,58.25,43.12,132.74,23.48,85.17,144.88,
     7129.85,159.95,69.61,117.77,96.03,113.72,72.84,50.77,99.64,110.1C,
     857.32,142.73,133.97,99.15,72.66,25.81,62.78,138.25,149.36,110.63,
     979.96,118.03,99.02,112.78,97.30,76.72,125.31,83.82,79.99,106.15,
     097.39,99.82,97.12,117.31,96.62,98.14,134.56,132.33,96.09,136.18,
     197.00,109.69,106.36,125.50,114.52,77.13,77.33,136.71,116.13,98.56,
     283.74,52.33,126.97,66.69,160.43,96.35,115.53,96.52,95.91,70.38,
     358.75,68.62,62.84,163.20,128.50,91.29,56.95,87.97,128.78,117.85,
     478.95,147.77,126.33,96.79,33.65,99.92,53.98,79.71,120.83,88.17,
     595.37,38.13/

          NAME=NAMES(I)
          L=LD(I)
      PA=A(J,I)*10.D**AX(J,I)
      PB=B(J,I)*10.D**BX(J,I)
      PC=C(J,I)*10.D**CX(J,I)
      PD=D(J,I)*10.D**DX(J,I)
      PE=E(J,I)*10.D**EX(J,I)
      PF=F(J,I)*10.D**FX(J,I)
      RETURN
      END

@PRT.S A.WBSIN
```

```
SH•COST(1).WBSIN
      1             SUBROUTINE WBSIN
      2             INCLUDE PARS
      3             INTEGER TYPE
      4       1     CONTINUE
      5             CALL SBLNK
      6             CALL PAGES(99)
      7             WRITE(6,110)
      8       30    READ(5,40) IBLK,CODE,TITLE,(ISUB(I1),I1=2,K1)
      9       40    FORMAT(I3,A5,6A4,16I3)
     10             IF(IBLK.EQ.0) RETURN
     11             IF(IBLK.GT.IS17.AND.IBLK.LT.IC1.OR.IBLK.GT.IC16) GO TO 50
     12             ISUB(1)=IBLK
     13       50    WRITE(6,25) IBLK,CODE,TITLE
     14             WRITE(8'IBLK,ERR=600) ISUB,CODE,TITLE
     15             GO TO 30
     16       600    WRITE(6,601)
     17       601    FORMAT(5X,'ERROR IN WBSIN')
     18        110  FORMAT(2X,'WBS DICTIONARY',///)
     19         25  FORMAT(5X,I5,2X,A5,2X,6A4)
     20             RETURN
     21       C      DEBUG INIT,SUBTRACE,TRACE
     22       C      AT 1
     23       C      TRACE ON
     24             END
```

140

```
@PRT,TL A.

SH•COST(1)  ELEMENT TABLE

0  NAME          VERSION        TYPE          DATE        TIME      SEQ #  SIZE-PRE,TEXT (CYCLE WORD) PSRMODE  LOCATION
   SALVAG                       FOR SYMB   19 DEC 78  12:30:37      1              7     5   13    5            1792
   LIFCYC                       ELT SYMB   21 DEC 78  15:50:30      2             35     5   47    5            1799
   SKALE                        FOR SYMB   30 JAN 79  09:32:05      3              2     5    0    1            1838
   NETREP                       FOR SYMB   30 JAN 79  12:05:19      4             36     5   35    5            1836
   SBLNK                        FOR SYMB   14 FEB 79  12:55:51      5              7     5    5    5            1872
   CALLID                       FOR SYMB   08 MAR 79  14:42:13      6              6     5    0    1            1879
   LEGEND                       ELT SYMB   23 MAR 79  07:51:07      7              9     5   22    5            1885
   PLOTS                        ELT SYMB   11 APR 79  09:29:57      8             14     5   46    5            1894
   SAVE                         ELT SYMB   02 JUL 79  11:52:18      9             18     5    0    1            1908
   DEPINC                       FOR SYMB   05 JUL 79  11:38:00     10             27     5   36    5            1926
   TOTALS                       FOR SYMB   05 JUL 79  11:39:28     11             15     5   46    5            1953
   SOLCST                       FOR SYMB   21 DEC 78  13:55:04     12             14     5   21    5            1968
   ENERGY                       FOR SYMB   18 JUL 79  13:11:36     13             18     5   45    5            1982
   POSAVE                       ELT SYMB   23 JUL 79  07:58:25     14             18     5   26    5            2000
   PLTURV                       ELT SYMB   15 AUG 79  09:26:48     15             24     5   15    5            2018
   PLTDRV                       RELOCATABLE 15 AUG 79 09:27:28     16      2      50                 QFR      2042
   LEGEND                       RELOCATABLE 15 AUG 79 09:58:33     17      2      14                 QFR      2098
   CALLID                       RELOCATABLE 15 AUG 79 10:03:27     18      2      13                 QFR      2110
   SUNRIZ                       FLT SYMB   17 AUG 79  09:18:09     19             21     5   24    5            2125
   SUNRIZ                       RELOCATABLE 17 AUG 79 09:22:31     20      2      32                 QFR      2145
   SFL                          FLT SYMB   04 SEP 79  08:59:24     21              8     5   13    5            2179
   GO                           ELT SYMB   19 SEP 79  08:55:03     22              9     5    7    5            2187
   TABLE                        ELT SYMB   19 SEP 79  08:58:13     23            208     5    2    3            2196
   TABLE                        RELOCATABLE 19 SEP 79 09:00:19     24      2     831                 QFR      2808
```

	SOLCST		RELOCATABLE	19 SEP 79	09:02:47	25	3	8				QFR 2837
	DEPINC		RELOCATABLE	19 SEP 79	09:06:40	26	3	30				QFR 2848
	ENERSY		RELOCATABLE	19 SEP 79	09:07:06	27	3	18				QFR 2881
	NETREP		RELOCATABLE	19 SEP 79	09:07:34	28	3	33				QFR 2902
	SALVAG		RELOCATABLE	19 SEP 79	09:09:36	29	3	6				QFR 2938
	SBLNK		RELOCATABLE	19 SEP 79	09:10:07	30	2	5				QFR 2947
	TOTALS		RELOCATABLE	19 SEP 79	09:11:29	31	3	15				QFR 2954
	LIFCYC		RELOCATABLE	19 SEP 79	09:12:33	32	3	22				QFR 2972
	POSAVE		RELOCATABLE	19 SEP 79	09:12:58	33	4	25				QFR 2997
	SFL		RELOCATABLE	19 SEP 79	09:13:35	34	3	9				QFR 3026
	PLOTS		RELOCATABLE	19 SEP 79	09:14:12	35	2	29				QFR 3038
	SKALE		RELOCATABLE	19 SEP 79	09:14:41	36	1	3				QFR 3069
•	SHMAP		MAP SYMB	19 SEP 79	09:26:48	37		2	5	4	5	3073
	CYPHER		ELT SYMB	19 SEP 79	11:41:51	38		15	5	20	5	3075
	CYPHER		RELOCATABLE	19 SEP 79	11:42:35	39	2	17				QFR 3090
	PARS		FOR PROC	20 SEP 79	10:20:07	40		31	5	11	5	3109
	LOAD		ELT SYMB	20 SEP 79	13:46:25	41		14	5	14	5	3140
	LOAD		RELOCATABLE	20 SEP 79	13:47:04	42	3	19				QFR 3154
	TEST1		ELT SYMB	20 SEP 79	14:23:21	43		21	5	6	5	3176
	TEST2		ELT SYMB	20 SEP 79	14:24:07	44		18	5	1	2	3197
	IFPRNT		FOR SYMB	21 SEP 79	07:58:42	45		31	5	53	5	3215
	IFPRNT		RELOCATABLE	21 SEP 79	07:59:12	46	3	34				QFR 3246
	TAXCST		FOR SYMB	21 SEP 79	08:12:20	47		28	5	47	5	3283
	TAXCST		RELOCATABLE	21 SEP 79	08:12:40	48	3	29				QFR 3311
•	PILE		ELT SYMB	21 SEP 79	11:40:18	49		8	5	19	5	3343
	RFPRNT		FOR SYMB	21 SEP 79	12:02:00	50		36	5	22	5	3351
	RFPRNT		RELOCATABLE	21 SEP 79	12:02:24	51	3	64				QFR 3387
	CSTDTO		FOR SYMB	21 SEP 79	14:25:18	52		53	5	61	5	3454
	CSTDTO		RELOCATABLE	21 SEP 79	14:25:59	53	3	89				QFR 3507
	PAGES		ELT SYMB	21 SEP 79	14:26:51	54		12	5	17	5	3599
	PAGES		RELOCATABLE	21 SEP 79	14:27:07	55	2	15				QFR 3611
	C		ABSOLUTE	21 SEP 79	14:40:28	56		68				QFR 3628
	AMORTZ		FOR SYMB	24 SEP 79	09:10:52	57		9	5	29	5	3696
	AMORTZ		RELOCATABLE	24 SEP 79	09:11:13	58	3	11				QFR 3705
	NFLATE		FOR SYMB	24 SEP 79	09:40:33	59		28	5	56	5	3719
	NFLATE		RELOCATABLE	24 SEP 79	09:40:43	60	3	36				QFR 3747
	CONCST		FOR SYMB	24 SEP 79	10:08:45	61		13	5	21	5	3786
	CONCST		RELOCATABLE	24 SEP 79	10:09:15	62	3	5				QFR 3799
	PVAC		FOR SYMB	24 SEP 79	10:09:54	63		18	5	62	5	3807
	PVAC		RELOCATABLE	24 SEP 79	10:10:23	64	3	17				QFR 3825
	RESULT		ELT SYMB	24 SEP 79	11:15:46	65		26	5	20	5	3845
	RESULT		RELOCATABLE	24 SEP 79	11:16:02	66	5	34				QFR 3871
	WBSIN		FOR SYMB	24 SEP 79	11:16:31	67		7	5	40	5	3910
	WBSIN		RELOCATABLE	24 SEP 79	11:17:06	68	3	8				QFR 3917
	TEST3		ELT SYMB	24 SEP 79	11:28:43	69		5	5	2	3	3928
	HANSMAN		ELT SYMB	24 SEP 79	11:48:50	70		40	5	1	2	3933
	CSTDTN		ELT SYMB	27 SEP 79	07:24:43	71		27	5	45	5	3973
	CSTDTN		RELOCATABLE	27 SEP 79	07:25:00	72	3	24				QFR 4000
	DIVVY		ELT SYMB	27 SEP 79	07:27:45	73		25	5	17	5	4027
	DIVVY		RELOCATABLE	27 SEP 79	07:27:59	74	3	30				QFR 4052
	SAMPLE	PROBLEM	ELT SYMB	27 SEP 79	09:14:48	75		7	5	11	5	4085
	SOLMAN		ELT SYMB	28 SEP 79	11:29:36	76		24	5	60	5	4092
	SOLMAN		RELOCATABLE	28 SEP 79	11:30:01	77	5	14				QFR 4116
•	SHCOST		ABSOLUTE	28 SEP 79	11:36:09	78		1051				QFR 4135
•	SHCOST		ELT SYMB	28 SEP 79	12:39:49	79		2	5	3	1	5186
	COST		ELT SYMB	28 SEP 79	12:39:57	80		2	5	4	1	5188

•	SHCOST	RELOCATABLE	28 SEP 79	12:33:29	81	2	4				QFR	5190
	MAP	ELT SYMB	28 SEP 79	12:34:04	82		1	5	0	1		5196
•	SHCOST	ELT SYMB	28 SEP 79	12:34:20	83		2	5	1	2		5197
•	SHCOST	RELOCATABLE	28 SEP 79	12:34:39	84	2	4				QFR	5199
•	SHCOST	ABSOLUTE	28 SEP 79	12:34:40	85		71				QFR	5205
•	BEGIN	ELT SYMB	28 SEP 79	12:35:08	86		1	5	0	1		5276
•	BEGIN	ELT SYMB	28 SEP 79	12:36:08	87		1	5	1	2		5277
•	BEGIN	ELT SYMB	28 SEP 79	12:38:03	88		1	5	2	3		5278
•	SHCOST	ELT SYMB	28 SEP 79	12:38:16	89		3	5	2	3		5279
•	SHCOST	RELOCATABLE	28 SEP 79	12:38:29	90	2	5				QFR	5282
•	SHCOST	ABSOLUTE	28 SEP 79	12:38:31	91		71				QFR	5289
	BEGIN	ELT SYMB	28 SEP 79	12:40:07	92		1	5	3	4		5360
•	SHCOST	ELT SYMB	28 SEP 79	12:41:33	93		3	5	3	4		5361
•	SHMAP	MAP SYMB	28 SEP 79	12:41:25	94		2	5	5	5		5364
•	SHCOST	ELT SYMB	28 SEP 79	12:42:05	95		3	5	4	5		5366
•	SHMAP	MAP SYMB	28 SEP 79	12:42:51	96		2	5	6	5		5369
	SHMAP	MAP SYMB	28 SEP 79	12:43:38	97		2	5	7	5		5371
	COST	ABSOLUTE	28 SEP 79	12:43:42	98		1351				QFR	5373
•	SHCOST	ELT SYMB	28 SEP 79	12:47:25	99		3	5	5	5		6424
•	SHCOST	ABSOLUTE	28 SEP 79	12:47:46	100		71				QFR	6427
•	SHCOST	ELT SYMB	28 SEP 79	12:48:10	101		3	5	6	5		6498
	SHCOST	ELT SYMB	28 SEP 79	12:48:41	102		4	5	7	5		6501
	SHCOST	RELOCATABLE	28 SEP 79	12:48:53	103	2	6				QFR	6505
	SHCOST	ABSOLUTE	28 SEP 79	12:48:55	104		72				QFR	6513
	PILE	ELT SYMB	28 SEP 79	12:50:24	105		6	5	20	5		6585

NEXT AVAILABLE LOCATION- 6591

ASSEMBLER PROCEDURE TABLE EMPTY

COBOL PROCEDURE TABLE EMPTY

FORTRAN PROCEDURE TABLE

D NAME	LOCATION	LINK	D NAME	LOCATION	LINK	D NAME	LOCATION	LINK
PARS	87C54	40						

ENTRY POINT TABLE EMPTY

&HOS,N

&BRKPT PRINTS